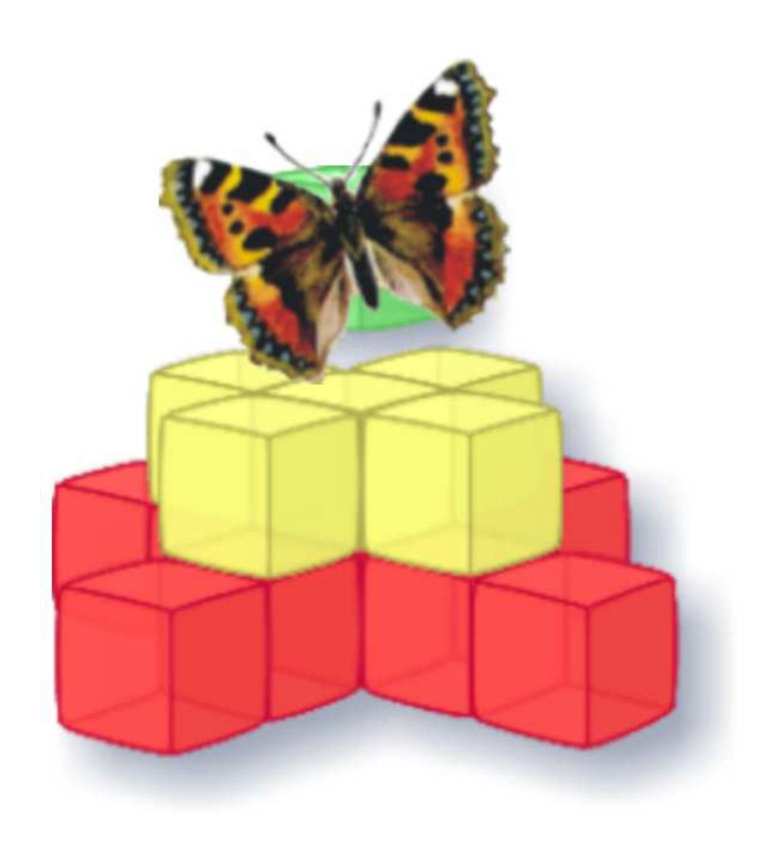
Dipl. Ing. Toralf Riedel
Dipl. Ing. Päd. Alexander Huwaldt

Benutzerhandbuch
SiSy® ARM
gültig ab SiSy-Version 3.30



1. Auflage: Oktober 2012

www.laser-co.de
www.myavr.de
service@myavr.de
Tel: ++49 (0) 3585 470 222
Fax: ++49 (0) 3585 470 233

Inhalt

1 Einleitung

Sie haben eine Ausgabe des Modellierungswerkzeuges Simple System, kurz SiSy, erworben. Bevor auf die verschiedenen Funktionen des Programms eingegangen wird, noch einige Worte zum vorliegenden Handbuch. Mit Hilfe des Handbuchs werden dem Nutzer die Grundlagen der Bedienung von SiSy erläutert. Der Inhalt, die Gestalt und die Regeln der Modelle werden nur am Rand betrachtet. Das genaue Vorgehen für die Programmierung eines Mikroprozessors wird an einem Beispiel ausführlich beschrieben. Auf die Grundlagen der Mikroprozessorprogrammierung wird im Rahmen dieses Handbuches nicht eingegangen. Dazu dienen die myAVR Lehrbücher.

Dem Nutzer wird in diesem Handbuch der Einstieg in das Programm erleichtert und die umfangreichen Funktionen von SiSy kurz und verständlich beschrieben. Bei der Arbeit mit SiSy erstellt der Anwender Modelle in Form von Diagrammen und in ihnen enthaltene Symbole. Die Grundlagen der Entstehung und Bearbeitung solcher Diagramme sind Gegenstand der Betrachtung dieses Handbuchs.

Folgende Darstellungs- und Gestaltungsmittel sind für den Nutzer bei der Arbeit mit SiSy besonders wichtig:

- die Diagramme als Fenster zur Ansicht und Bearbeitung von Modellen;
- der Navigator als Fenster zur Steuerung und Bewegung in Modellen;
- der Assistent mit Hilfestellungen zum jeweils geöffneten Diagramm und mit Diagrammvorlagen (wenn vorhanden);
- die Menüs und Schalter für Befehle an Navigator, Diagramm und Objekt im Kontext mit der Modellierung.

Zu den Bezeichnungen im Text:

- Falls ein Menübefehl nur über Untermenüs zu erreichen ist, werden die einzelnen Menübezeichnungen kursiv geschrieben und durch Schrägstriche voneinander getrennt.
  Beispiel: Menü *Hilfe/über SiSy*

- Titel von Dialogboxen, Schaltflächen und Menüpunkten werden in Anführungszeichen gesetzt.
  Beispiel: Dialogbox „Definition“, Schaltfläche „OK“

2 Vorbereitung

In diesem Kapitel werden Sie über notwendige Schritte zur Installation, Konfiguration und Aufbau einer funktionsfähigen Entwicklungsumgebung informiert.

2.1 Installation der Software

Für die Bearbeitung der Übungen und Aufgaben steht Ihnen die Entwicklungsumgebung SiSy AVR, SiSy AVR++ bzw. SiSy ARM zur Verfügung. Sollten Sie SiSy bereits installiert haben, können Sie dieses Kapitel überspringen.

2.1.1 Voraussetzungen

Für die Installation benötigen Sie einen Freischaltcode (Lizenzangaben). Falls Sie diese Angaben nicht mit der Software erhalten haben, können Sie diese online abrufen von

www.myAVR.de → Online-Shop→ Kontakt/Service

oder fordern Sie diese beim Hersteller an:

Tel: 03585-470222
Fax: 03585-470233
e-Mail: support@myAVR.de.

Bitte prüfen Sie, ob die Systemvoraussetzungen für die Installation und die Arbeit mit SiSy AVR / SiSy ARM für die Mikrocontrollerprogrammierung gewährleistet sind.

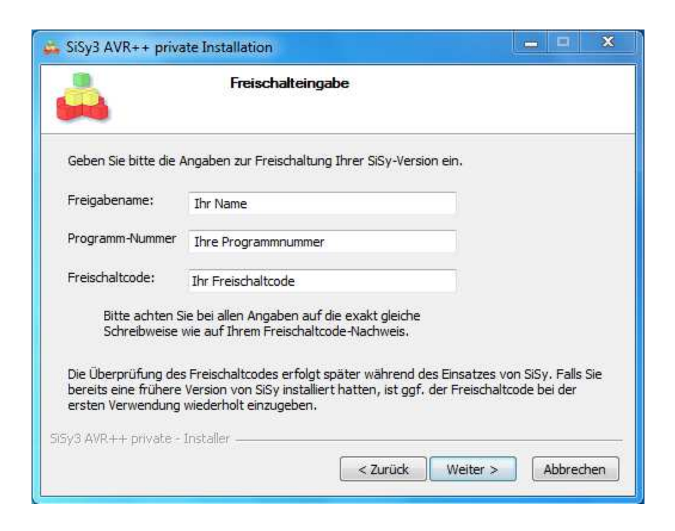
- für das Board myAVR Board MK2, myAVR Board MK3 und STM32F4-Discovery einen PC-Arbeitsplatz oder Notebook mit USB-Anschluss
- für das myAVR Board MK1 einen PC-Arbeitsplatz oder Notebook mit LPT-Port und mindestens einem COM-Port
- PC mit Windows XP, Vista oder Windows 7
- Microsoft Internet-Explorer 7 oder höher
- Maus oder ähnliches Zeigegerät
- Assembler bzw. C/C++ Entwicklungsumgebung (in SiSy bereits integriert)
- myAVR Board bzw. STM32F4-Discovery
- Programmierkabel
  - USB Kabel (myAVR Board MK2)
  - Mini-USB-Kabel (STM32F4-Discovery)
  - LPT-Verlängerung und Null-Modemkabel (myAVR Board MK1)
- Bei Bedarf (z.B. autonomer Einsatz des Boards) geeignete Spannungsversorgung z.B. 9 V Batterie / stabilisiertes 9 V Netzteil

Des Weiteren sollten Sie Grundkenntnisse in einer beliebigen Programmiersprache besitzen. Die Installation und der erste Start müssen mit Administratorrechten durchgeführt werden.

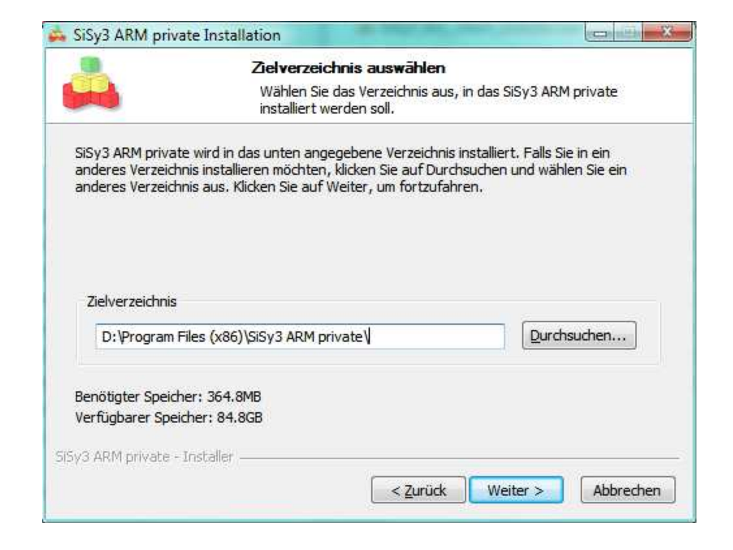
2.1.2 Setup von der SiSy-CD

Legen Sie die CD „SiSy" in Ihr CD-ROM-Laufwerk ein. Falls die CD nicht automatisch startet, wählen Sie bitte im Explorer das CD-ROM-Laufwerk und starten die setup.exe aus dem Pfad *CD-Laufwerk:\Ausgabe\SiSy*.
Auf dem Startbildschirm stehen Schaltflächen zur Verfügung zum Installieren der Software und zum Öffnen von Begleitdokumenten.
Für die Installation der Software betätigen Sie die entsprechende Schaltfläche. In Abhängigkeit Ihrer Rechnerkonfiguration kann der Start des Setup-Programms einige Sekunden dauern. Das gestartete Setup-Programm wird Sie durch die weitere Installation führen.

Beginn der Installation
Betätigen Sie im Setup-Programm die Schaltfläche „Weiter“. Sie erhalten die Lizenzbestimmungen. Bitte lesen Sie diese sorgfältig durch. Wenn Sie sich mit diesen Bestimmungen einverstanden erklären, bestätigen Sie die Lizenzbestimmungen mit der Schaltfläche „Annehmen“.
Sie werden im folgenden Dialog dazu aufgefordert, Ihre Lizenzangaben einzugeben.



Danach erscheint die Dialogbox „Komponenten auswählen“, welche Sie mit „Weiter“ bestätigen. Im darauf folgenden Fenster können Sie festlegen, unter welchem Pfad SiSy AVR / SiSy ARM installiert werden soll. Wenn ein anderer Pfad (bzw. ein anderes Laufwerk) gewünscht wird, aktivieren Sie die Schaltfläche „Durchsuchen“. Eine Dialogbox erscheint, in der Sie Laufwerk und Verzeichnis auswählen können.

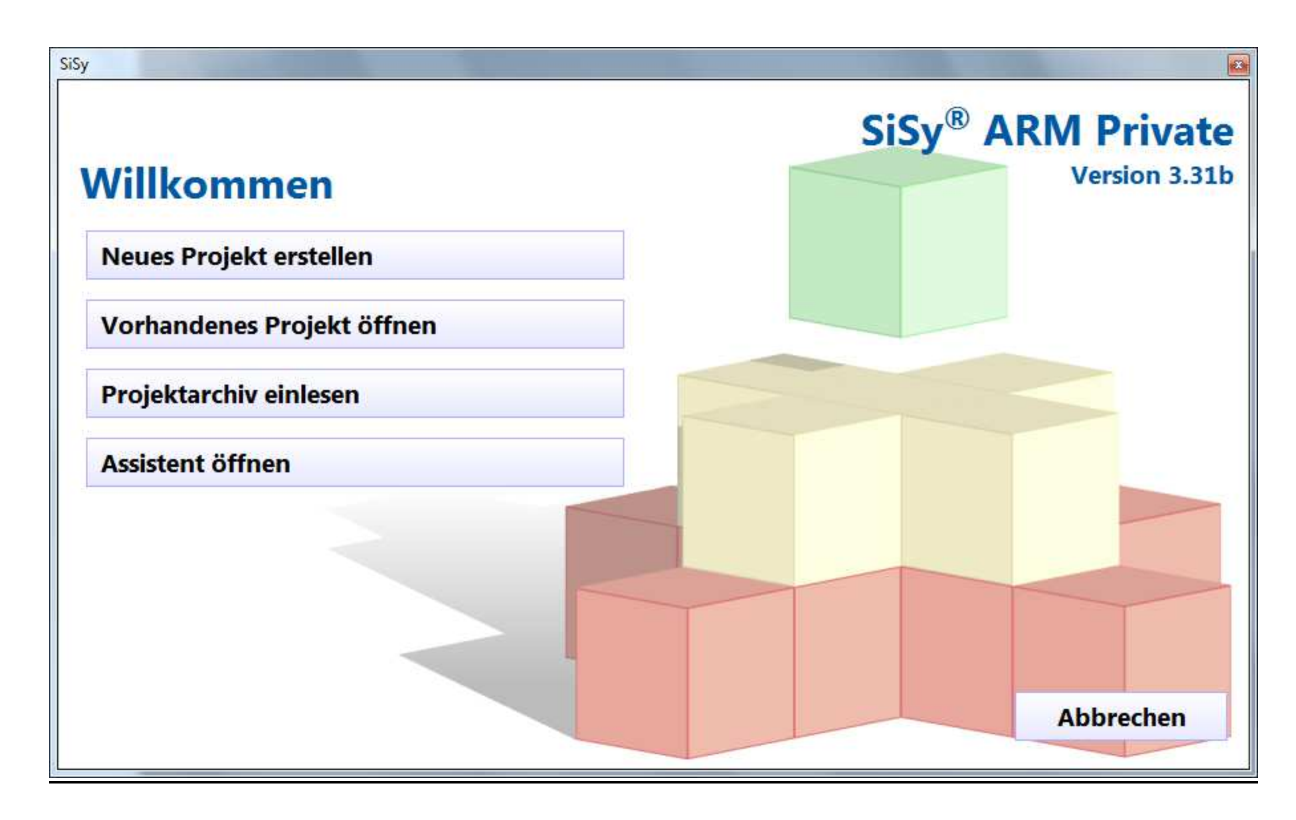




Bestimmen Sie danach den Startmenü-Ordner, in dem die Verknüpfungen von SiSy eingefügt werden. Sie können den Zielordner ändern; Sie können dies durch auswählen von „Keine Verknüpfungen erstellen“ unterbinden.

Beginnen Sie nun die Installation durch Betätigen der Schaltfläche „Installieren“. Die Installation wird nach aktivieren von „Fertig stellen“ abgeschlossen.

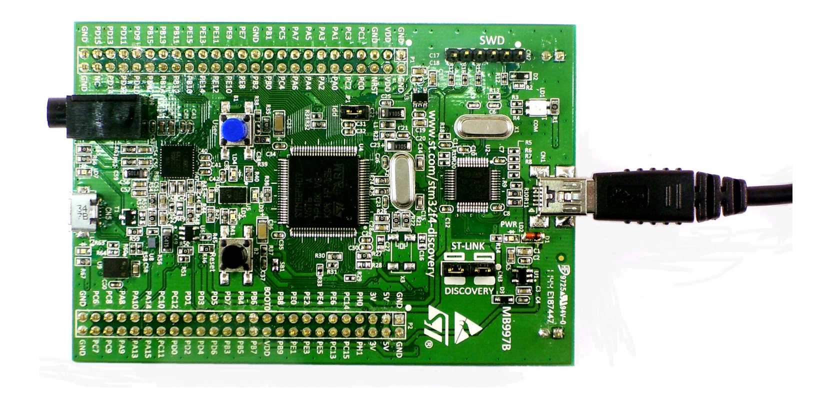
Sie können nun SiSy starten. Es erscheint auf Ihrem Bildschirm der Dialog „Willkommen in SiSy“. Folgen Sie dann den Hinweisen des Assistenten, indem Sie „Assistent öffnen“ auswählen.



Hinweis:
In SiSy sind 2 Dateien enthalten, die Makros beinhalten („handbuch.doc“, „multi.doc"). Von einigen Virenscannern werden diese Makros als „Virus“ erkannt und entsprechend behandelt. In den Heuristik-Einstellungen des Virenscanners kann diese Behandlung unterdrückt werden.

2.2 Beschaffen bzw. Herstellen der Hardware

Alle Ausführungen, Übungen und Aufgabenstellungen in diesem Benutzerhandbuch beziehen sich auf ein myAVR Board als Referenzhardware. Für die Arbeit mit STM32F4-Discovery und SiSy ARM steht ein separates Handbuch zur Verfügung.
Wenn Sie Spaß an Elektronik haben, können Sie die myAVR Boards auch selbst fertigen. Die Komponenten erhalten Sie als Bausatz inklusive Schaltplan. Dokumente sowie Anwendungsbeispiele zu den myAVR Boards stehen unter www.myAVR.de zum Download bereit.
Ausführliche Beschreibungen zur Programmierung mit Assembler, C/C++, BASCOM sind nicht Inhalt dieses Handbuches. Weiterführende Erklärungen dazu sind im „myAVR Lehrbuch Mikrocontrollerprogrammierung“ enthalten.



STM32F4-Discovery mit Mini-USB-Kabel

3 Arbeiten mit SiSy, allgemein

Um dem Nutzer zu erläutern, wie er in SiSy modellieren kann, werden zweckentsprechende Definitionen der Begriffe gegeben.

3.1 Projektarbeit

3.1.1 Was ist ein SiSy-Projekt?

Ein SiSy-Projekt ist eine abgegrenzte Menge von verknüpften Elementen für ein zu bearbeitendes Problem. Alle aktuellen Daten sind in einer Projektdatenbank gespeichert. Die Projektdatenbank besteht aus einer Anzahl von Dateien im Projektverzeichnis, wobei jedes Projekt sein eigenes Verzeichnis hat. Durch das Anlegen eines Projektarchivs können diese in einer Datei komprimiert werden.

3.1.2 Neues Projekt erstellen

Ein neues Projekt wird eingerichtet und für die Bearbeitung bereitgestellt. Die Definition (Erstellung) eines neuen Projektes erfolgt durch Vergabe eines Projektdateinamens und/oder durch Überschreiben eines alten Projektes gleichen Namens. Über die Schaltfläche „Ordner für Projekte ändern“ kann die vorgeschlagene Pfadangabe geändert werden.

<u>*Hinweis:*</u>

Falls ein bereits vorhandener Projekt- oder Verzeichnisname gewählt wird, erscheint folgende Meldung:

> „Achtung: ein Projekt dieses Namens existiert bereits. Ändern Sie den Projektnamen oder das vorhandene Projekt wird gelöscht und durch das neue ersetzt.“

Nach der Vergabe eines Projektnamens kann im nachfolgenden Fenster ein Vorgehensmodell ausgewählt werden. Je nach Vorgehensmodell können Vorlagen oder Assistenten das Erstellen eines neuen Projektes unterstützen.

3.1.3 Vorhandenes Projekt öffnen

Ein vorhandenes Projekt wird geöffnet und die abgespeicherten Diagramme und Objekte sowie alle Möglichkeiten für die weitere Bearbeitung werden verfügbar gemacht. Die Wahl des Projektes erfolgt durch Klicken auf den entsprechenden Namen oder über die Schaltfläche „Projekt suchen“.

Ist der Mauszeiger auf dem Projektnamen oder auf den Symbolen, dann werden Informationen angezeigt.

3.1.4 Projekt archivieren

Menü *Projekt/Archiv/Anlegen.*

Es kann ein komprimiertes Archiv des Projektes erzeugt werden. Dies ist besonders aus Gründen der Datensicherheit sinnvoll. Zielverzeichnis und Dateiname für die Archivdatei werden vorgeschlagen und können korrigiert werden. Beim Aktivieren der Checkbox „Datum anfügen“ wird das Projekt mit Änderungsdatum gespeichert. Wenn ein Projekt unter einem bereits vorhandenen Archivnamen angelegt werden soll, wird eine Warnung vor dem Überschreiben angezeigt. Bei Auswahl von „Nein“ wird die Erstellung des Archivs abgebrochen, bei „Ja“ wird das Projekt archiviert.

<u>*Hinweis:*</u> SiSy bietet die Möglichkeit des regelmäßigen Abspeicherns verschiedener Arbeitsstände, d.h. ein archiviertes Projekt wird nicht überschrieben. Ein Projektstand kann in einer neuen Archivdatei abgelegt werden.

3.1.5 Projektarchiv einlesen

Menü *Projekt/Archiv/Einlesen*.

Hierunter versteht man das Einlesen eines Archivs zum Zweck der Rekonstruktion des Projektes.

Einlesen bedeutet Entpacken eines archivierten Projektes. Dazu muss das Archiv ausgewählt sein. Beim Aktivieren der Checkbox „Datum entfernen“ wird das Datum nicht im Projektnamen geschrieben.

Hinweis:

Wenn im Zielpfad des Entpackens bereits ein Projekt existiert, erscheint eine Überschreibwarnung.

3.1.6 Projektarchiv als Email versenden

Menü *Projekt/Archiv/Als Email versenden…*.

Um Projekte per Email zu versenden, sollten diese zu einem Archiv zusammengefasst werden. Die Funktion „Projektarchiv als Email versenden“ erzeugt von dem aktuell geöffneten Projekt ein Archiv, startet das Standard-Email-Programm Ihres Systems, legt eine neue Email an und fügt das Archiv als Anlage ein. Sie können jetzt die Email schreiben und sofort das Projektarchiv senden.

Hinweis:

Das Standard-Email-Programm Ihres Systems muss ordnungsgemäß eingerichtet und registriert sein, damit diese Funktion korrekt ausgeführt werden kann. Es ist nicht möglich, diese Funktion mit einem WEB-Browser-Email-Programm zu nutzen.

3.2 Die Modellierungselemente von SiSy

Werkzeug

SiSy stellt für die Bearbeitung der Modelle und Teilmodelle Werkzeuge der entsprechenden Methodik bereit. Werkzeuge sind Editoren, mit denen in einem Fenster die grafische Darstellung (Diagramme) der Modelle bearbeitet werden kann.

Modellelemente

Modelle bestehen aus Elementen, welche zueinander in Beziehung stehen. Diese Modellelemente werden nach bestimmten Regeln (Metamodell) durch SiSy verarbeitet und dargestellt (Notation). Modellelemente sind in der Regel Knoten (Objekte), Kanten (Verbindungen), Rahmen, Referenzen auf Modellelemente und Texte. Prinzipiell kann jedes Modellelement verfeinert werden. Typisch ist es Knoten (Objekte) dadurch zu verfeinern, dass „unter“ diesen Objekten weitere Modellelemente angelegt werden. Dabei entsteht eine Modellhierarchie. Jedes Modellelement verfügt über einen Kurznamen, einen Langnamen, eine Beschreibung und modellinterne Attribute (Inhalt).

Diagramme

Diagramme sind grafische Darstellungen von Modellen oder Teilmodellen, die mit einem bestimmten Werkzeug erstellt werden. Die Modellierungselemente werden als Objekte in den Diagrammen unter Einhaltung von Regeln zusammengestellt.

Objekte

Objekte sind mögliche Modellelemente in Diagrammen, z.B. „kleines Programm“ in der „Programmierung“. Objekttypen sind konkrete Ausprägungen von Objekten, die in einem Diagramm angelegt wurden, z.B. Objekttyp „Lauflicht“ vom Objekt „kleines Programm“.

Referenzen

Die Objekte eines Diagramms können in anderen Diagrammen wieder verwendet werden. Durch das Hineinziehen aus dem Navigator oder aus einem offenen Diagramm wird eine Referenz vom Originalobjekt erzeugt. Die Referenz ist nur ein Verweis auf das Original, alle angezeigten Informationen wie Kurzname, Langname und Beschreibung werden vom Original bezogen. Somit sind die Informationen in allen Referenzen eines Objektes identisch mit dem Original. Dadurch werden Änderungen dieser Informationen automatisch auf alle Referenzen übertragen. Weiterhin ist es möglich, diese Referenzierung über einen so genannten Report auszuwerten.

Kanten

Kanten sind Verbindungselemente zwischen Objekten. Eine Verbindung wird durch Ziehen mit der Maus (linke Maustaste) vom Verteiler des selektierten Objektes auf das gewünschte Objekt erreicht. Nach Loslassen der Maustaste und Prüfung der Verbindungszulässigkeit durch SiSy erscheint ein Kanten-Dialog, in dem das Element definiert und individuelle Einstellungen getroffen werden können.

Hinweis:

Bei Verbindung mit gehaltener STRG-Taste wird die Prüfung vernachlässigt und eine Zwangsverbindung erreicht.

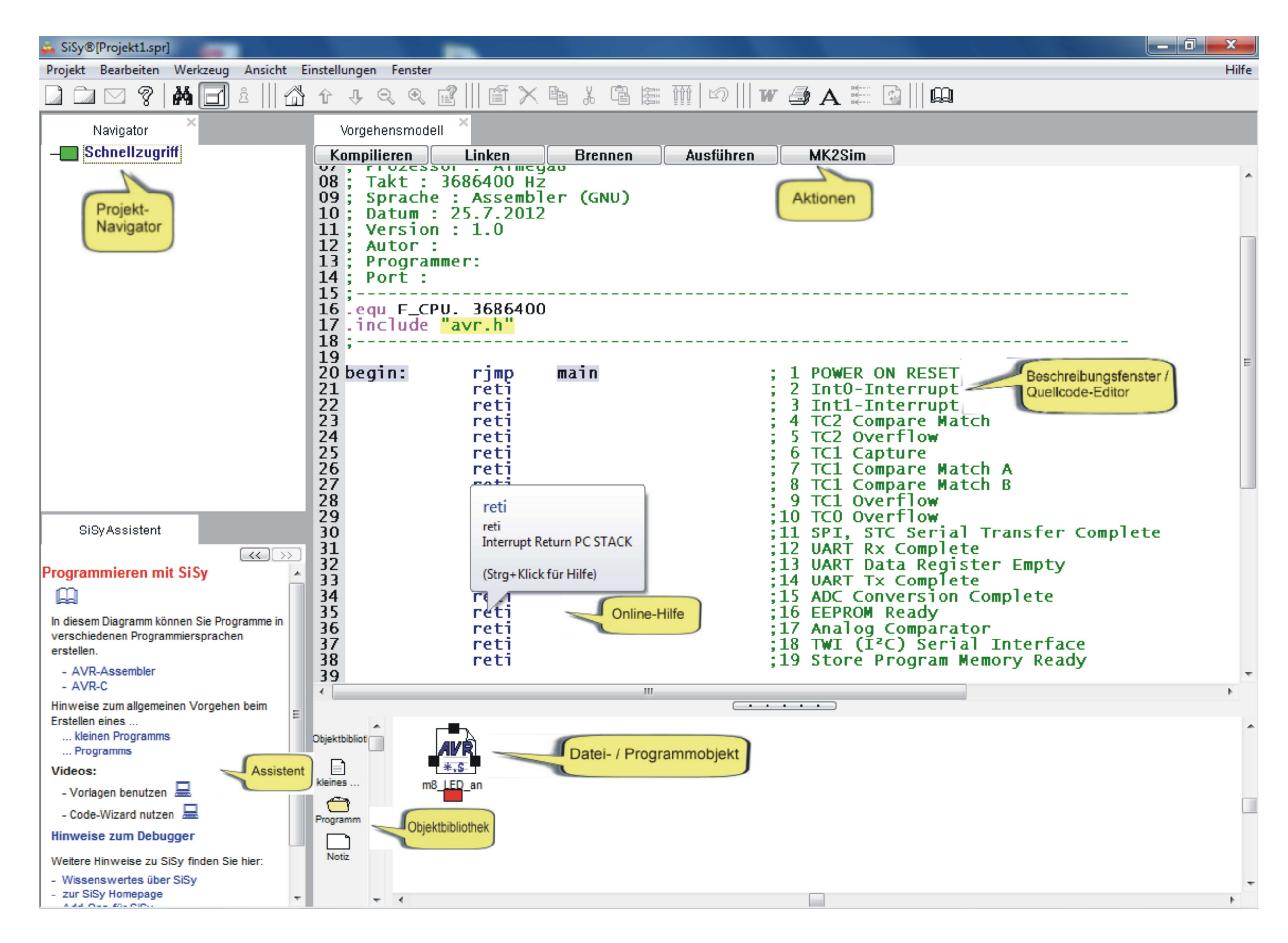
Rahmen

Ein Rahmen fasst ausgewählte Objekte des Diagramms optisch zusammen. Er besitzt einen Kurz- sowie Langnamen und eine Objektbeschreibung, kann also als Objekt aufgefasst werden.

Hinweis:

Inhalte von Rahmen sind in Reports oder einer Dokumentengenerierung auswertbar.

3.3 Die Fenster für die Modellierung



Navigator

Dieser befindet sich im linken, oberen Teil des Fensters. Er ermöglicht dem Anwender die Orientierung über die in der Projektdatenbank gespeicherten Objekte sowie deren Bereitstellung für die weitere Verwendung. Nach dem Start von SiSy werden neben dem Vorgehensmodell eine Reihe leicht zu handhabender Schaltflächen, Menüs und weitere Hilfsmittel angezeigt.

SiSy Assistent

Am linken unteren Bildschirmrand befindet sich diese Nutzerunterstützung.

- Er begleitet den Anwender durch das gesamte Projekt und hält immer passende Informationen zur aktuellen Sicht parat
- Er ist auf die jeweilige Ausgabe von SiSy bezogen.
- Oft können Beispiele als Vorlagen vom Assistenten geladen werden.

Diagrammfenster

Am oberen Bildrand befinden sich das Hauptmenü, das hilfreiche Funktionen zu SiSy bereithält, und eine Werkzeugleiste, mit deren Hilfe schnell auf nützliche Anwendungen zugegriffen werden kann. Das Diagrammfenster nimmt die rechte Bildschirmseite ein und ist der Raum, in dem der Nutzer modelliert. Es enthält:

- das ausgewählte Vorgehensmodell,
- die Objektbibliothek mit den möglichen Objekten des aktuellen Diagramms sowie
- ein Fenster zur Beschreibung des markierten Objekts, in diesem Fall zum Editieren des Quelltextes.

Die Bedienelemente/Objektbibliothek

SiSy bietet, wie bei Windows-Anwendungen üblich, die Steuerung von Befehlen über das Hauptmenü, über die Werkzeugleisten, die Tastatur oder die Objektbibliothek an. Darüber hinaus enthalten das Kontextmenü und der Navigator Steuerfunktionen.

Die Anzahl der möglichen Befehle in der Menüleiste ist abhängig davon, ob ein Projekt geöffnet ist. Ist das nicht der Fall, erscheint ein Menü mit wenigen Befehlen. Bei einem geöffneten Projekt hält SiSy umfangreichere Menüs bereit. Die wichtigsten Menübefehle befinden sich auch als grafische Schaltfläche in der Werkzeugleiste, die eine schnelle und effiziente Bedienung des Programms ermöglicht. Die Toolbox-Darstellung bietet dem Anwender wichtige Programmfunktionen als direkten Link an.
Ein weiteres Bedienelement ist die Objektbibliothek. Sie unterstützt das Anlegen neuer Objekte.

3.4 Modelle bearbeiten

3.4.1 Modellelemente anlegen

Modellelemente können auf verschiedene Art und Weisen erzeugt werden. Der häufigste Weg ist das Anlegen über die Objektbibliothek. Dabei wird das gewünschte Element per Drag und Drop von der Objektbibliothek in das dazugehörige Diagrammfenster gezogen. Referenzen auf vorhandene Modellelemente werden üblicherweise dadurch erzeugt, dass diese per Drag und Drop aus dem Navigator in das gewünschte Diagramm gezogen werden. Verbindungen lassen sich am Einfachsten über den Verteiler herstellen. Sie können aber Modellelemente auch durch kopieren oder importieren anlegen.

3.4.2 Modellelemente auswählen

Die meisten Modellelemente lassen sich in Diagrammen auswählen (selektieren). Das Selektieren erfolgt in der Regel durch Anklicken (linke Maustaste) mit dem Mauscursor. Eine Selektion per Klick mit der rechten Maustaste ist ebenfalls möglich, dabei öffnet sich nach dem Auswählen sofort das Kontextmenü für das selektierte Modellelement. Das Betätigen der Tabulatortaste bewirkt das Auswählen des nächsten Modellelementes in der Reihenfolge des Anlegens der Elemente. Wird über die Auswahl mit der Tabulatortaste das letzte Element erreicht, beginnt die nächste Iteration wieder beim ersten Modellelement im Diagramm. Durch Betätigen der Umschalttaste kann die Iterationsrichtung umgeschaltet werden. Es ist ebenfalls möglich, mehrere Modellelemente auszuwählen. Dafür muss beim Selektieren mit der Maus die Umschalttaste gehalten werden. Bei gehaltener Umschalttaste lassen sich mehrere Objekte (keine Kanten und Rahmen) durch Ziehen einer Markise mit dem Mauscursor auswählen. Über den Menüpunkt *Bearbeiten/Diagramm/Alles markieren* können Sie alle Objekte des aktiven Diagramms selektieren.

3.4.3 Modellelemente untersuchen (Report)

Modellelemente sind in den meisten Fällen nicht isoliert sondern haben vielfältige Beziehungen zu anderen Modellelementen. Um bei Änderungen, wie zum Beispiel Umbenennen oder Löschen, mögliche Auswirkungen auf das gesamte Modell zu überblicken, bietet der Objektreport die Möglichkeit die Modellbeziehungen des ausgewählten Objektes auszuwerten. Wählen Sie ein Objekt aus und öffnen Sie den Reportdialog über das Hauptmenü, die Werkzeugleiste oder das Kontextmenü (diagrammabhängig). Wählen Sie die erforderliche Modellanfrage und starten den Report. Das Ergebnis der Modellanfrage (z.B. Alle angebundenen Objekte) wird als Liste angezeigt. Sie können dann zu diesen Modellelementen navigieren.

3.4.4 Modellelemente verschieben

Bestimmte Modellelemente (z.B. Knoten/Objekte) lassen sich im Diagramm frei anordnen. Das Positionieren erfolgt in der Regel durch Drag und Drop mit der Maus. Die Modellelemente werden dabei an einem Raster ausgerichtet. Das Raster erleichtert die Positionierung. Um die Rasterfangfunktion zu unterbinden, muss beim Drag und Drop eines Elementes die Umschalttaste betätigt werden. Viele Modellelemente lassen sich auch über die Cursortasten verschieben. Dabei bewegen sich die Modellelemente in Richtung der betätigten Taste auf dem Raster. Auch dabei lässt sich durch die Umschalttaste die Rasterfangfunktion unterbinden. Bei Kanten wird ein Stützpunkt verschoben, der in Abhängigkeit der ausgewählten Kantenform zum gewünschten Ergebnis der Darstellung führt.
Rahmen haben nur eine eigene Position, solange diese leer sind. Ein Rahmen bezieht sich in Position und Größe immer auf die darin enthaltenen Modellelemente (umschließen). Beim Verschieben eines Rahmens mit enthaltenen Elementen wird nicht der Rahmen, sondern die enthaltenen Modellelemente verschoben.

3.4.5 Die Größe von Modellelementen verändern

In manchen Diagrammen verfügen Modellelemente über Anfassmarken zur Änderung der Größe. Die manuelle Größenänderung von Modellelementen ist jedoch in vielen Diagrammtypen nicht erwünscht. In diesen Diagrammen stehen keine Anfassmarken zu Größenänderung zur Verfügung. Sollte in Ausnahmefällen eine manuelle Anpassung der Objektgröße gewünscht sein, kann dies für selektierte Objekte durch die Kombination von gehaltener Steuerungstaste und den Cursortasten bzw. die Tasten Plus, Minus und Mal des Numerikfeldes erfolgen.

3.4.6 Modellelemente verbinden

Knoten und Rahmen können über Kanten verbunden werden (Verbindungen). Verbindungen werden als Linien mit unterschiedlichen Linientypen und -formen sowie Symbolen (Anfang, Ende, Mitte) und diagrammspezifischen Beschriftungen dargestellt. Das Verbinden erfolgt in der Regel über den Verteiler (rotes Rechteck an der Unterkante) eines selektierten Objektes. Dabei wird per Drag und Drop vom Verteiler die Verbindung vom Quellelement zum Zielelement gezogen. Es ist nicht nötig exakt den Rand des Zielelementes zu treffen sondern besser, die Verbindung weit in das Zielobjekt hinein zu ziehen. Nach dem Loslassen der linken Maustaste wird die Verbindung erzeugt und die Randpunkte bis zu denen die Verbindung gezeichnet wird (Clipping) automatisch ermittelt. Sie können während des Drag und Drop durch Klick mit der rechten Maustaste bereits den zukünftigen Stützpunkt der Verbindung festlegen.

3.4.7 Modellelemente löschen

Zu löschende Modellelemente müssen zuerst selektiert werden. Die Löschaufforderung erfolgt mittels der Taste „Entfernen“, der Werkzeugleiste oder über das Hauptmenü *Bearbeiten/Objekt/Löschen* bzw. das Kontextmenü. Der Löschvorgang selbst ist abhängig von der Art des zu löschenden Modellelementes. Zum Beispiel werden frei stehende Referenzen ohne Rückfrage gelöscht, Originale werden in zwei Stufen gelöscht. Dabei erfolgt zuerst das Entfernen aus dem Diagramm und dann auf Anfrage auch das Löschen aus der Datenbank. Objekte die verfeinert oder verknüpft sind lassen sich nicht ohne erhöhten Aufwand aus dem Modell entfernen. Zuerst müssen die Modellbeziehungen des Elementes aufgelöst werden bis es für den Löschvorgang isoliert ist. Dazu springt in entsprechenden Fällen der Löschassistent an. Dieser gibt dem Anwender die Möglichkeit zuerst die Auswirkungen auf das Gesamtmodell zu beurteilen und kann dann auf Anforderung komplexe Löschvorgänge, zum Beispiel von ganzen Teilmodellen in der Modellhierarchie, durchführen.

3.4.8 Modellelemente kopieren

Um eine Kopie eines Modellelementes zu erzeugen wählen Sie im Menü oder in der Werkzeugleiste den Befehl „Kopieren“. Sie können dieses Objekt jetzt innerhalb des Projektes in ein Diagramm passenden Typs einfügen. Sollte das Diagramm den Objekttyp nicht akzeptieren, erhalten Sie dazu eine Meldung. Rahmenelemente wie zum Beispiel Klassen können mit deren Inhalt kopiert werden. Dabei werden jedoch Modellelemente in der Verfeinerung nach unten nicht berücksichtigt (flache Kopie). Innerhalb eines Diagramms können Sie von einzelnen Objekten durch Drag und Drop bei gleichzeitig gedrückter Steuerungstaste Kopien erzeugen.

Hinweis:

Beachten Sie, dass eine Kopie immer dem kopierten Element entspricht. Das heißt, die Kopie eines Originals ist wieder ein neues Original, die Kopie einer Referenz ist eine neue Referenz auf das ursprüngliche Original.

3.4.9 Modellelemente ausschneiden

Zum Ausschneiden eines Modellelements wählen Sie im Menü oder der Werkzeugleiste *Bearbeiten/Objekt/Ausschneiden*. Sie können dieses Objekt jetzt innerhalb des Projektes in ein Diagramm passenden Typs einfügen. Sollte das Diagramm den Objekttyp nicht akzeptieren, erhalten Sie dazu eine Meldung. Rahmenelemente wie zum Beispiel Klassen können mit deren Inhalt verschoben werden. Dabei werden auch Modellelemente in der Verfeinerung nach unten berücksichtigt.

Hinweis:

Beachten Sie, dass dieser Vorgang starke Auswirkungen auf die Modellstruktur hat. Vergewissern Sie sich über den Objektreport über die möglichen Auswirkungen.

3.4.10 Diagramme kopieren (flache Kopie)

Sie können Kopien von gesamten Diagrammen anlegen und einfügen. Dabei unterscheidet man zwischen einer flachen Kopie bei der alle Elemente der gezeigten Ebene (Diagramm) kopiert werden und einer tiefen Kopie bei der alle Elemente dieser Ebene und der darunter liegenden Ebenen (Teilhierarchie) kopiert werden. Eine flache Kopie erzeugen Sie über die Menübefehle *Bearbeiten/Diagramm/Diagrammkopie anlegen (flache Kopie)* und *Bearbeiten/Diagramm/Diagrammkopie einfügen (flache Kopie).*

Hinweis:

Beachten Sie, dass das Einfügen der Diagrammkopie nur in einem Diagramm des gleichen Typs und innerhalb des gleichen Projektes möglich ist.

3.4.11 Modelle kopieren (tiefe Kopie)

Sie können Kopien von gesamten Teilmodellen anlegen und einfügen. Dabei unterscheidet man zwischen einer flachen Kopie bei der alle Elemente der gezeigten Ebene (Diagramm) kopiert werden und einer tiefen Kopie bei der alle Elemente dieser Ebene und der darunter liegenden Ebenen (Teilhierarchie) kopiert werden. Eine tiefe Kopie erzeugen Sie über die Menübefehle *Bearbeiten/Diagramm/Diagrammkopie anlegen (tiefe Kopie)* und *Bearbeiten/Diagramm/Diagrammkopie einfügen (tiefe Kopie).*

Hinweis:

Beachten Sie, dass eine tiefe Kopie dem Wesen nach ein Export und Import von Teilmodellen darstellt. Das Einfügen eines solchen Imports in ein anderes Projekt kann zu Modellkonflikten führen.

3.4.12 Die Modellhierarchie bearbeiten (Jo-Jo)

Die Modellhierarchie kann durch Einfügen neuer Ebenen oder Auflösen von Modellebenen verändert werden (Jo-Jo). Diese Funktion erreichen Sie am besten über das Haupt- oder Kontextmenü. Um eine gewünschte Anzahl von Modellelementen zu einer neuen Ebene zusammenzufassen, selektieren Sie diese und wählen im Menü

Bearbeiten/Objekt/Jo-Jo/Objekte zusammenfassen. Es wird ein neues Objekt erzeugt und in das Diagramm eingefügt. Die ausgewählten Objekte wurden dem neuen Objekt untergeordnet. Bei der Funktion *Jo-Jo/Objekt auflösen* werden die untergeordneten Objekte in das aktuelle Diagramm eingefügt und das ausgewählte Objekt gelöscht.

3.5 Die Ansicht verändern

3.5.1 Fensterinhalt verschieben

Modelle werden oft viel größer als der sichtbare Bereich des Diagrammfensters. Um den sichtbaren Teil eines Diagramms zu verändern (Scroll) haben Sie folgende Möglichkeiten:

- die Scrollleisten des Fensters benutzen
- den Diagrammhintergrund per Drag und Drop schieben
- wenn kein Modellelement ausgewählt ist, mit den Cursortasten scrollen
- Menüpunkt *Ansicht/Fenstermitte* um zum Diagrammursprung zurück zu kehren

Hinweis:

Beachten Sie, dass der Diagrammursprung nicht wie bei einer Textverarbeitung oben links des Arbeitsbereiches liegt sondern wie bei einer MindMap oder vielen Konstruktionsprogrammen (CAD) in der Mitte (Nullpunkt, Fenstermitte) und das Modell sich somit in alle Richtungen entwickeln lässt. Vergleichen Sie dazu die Position der Scrollbalken. Versuchen Sie auf keinen Fall zu Beginn der Arbeit die Scrollposition oben links herzustellen!

3.5.2 Fensterinhalt vergrößern oder verkleinern

Um einen Überblick über größere Diagramme zu gewinnen, können Sie die Darstellung vergrößern oder verkleinern (Zoom). Eine elegante Möglichkeit das aktuelle Diagramm zu zoomen ist das Scrollrad der Maus. Sie können aber auch über das Kontextmenü oder den Menüpunkt *Ansicht* feste Zoom-Faktoren einstellen.
Die Funktion *Ansicht/Zoom/Alles* aus dem Menü oder der Werkzeugleiste passt das Diagramm an die Fenstergröße an. Eine weitere Möglichkeit das Diagramm zu zoomen, erhalten Sie über die Tasten des Numerikfeldes:

- Numerikfeld **+** ... Ansicht vergrößern
- Numerikfeld **-** ... Ansicht verkleinern
- Numerikfeld ***** ... Ansicht alles (einpassen)

3.5.3 Fensterinhalt farbig oder als Kontur darstellen

Temporär kann die farbige Darstellung von Elementen unterbunden werden. Dazu können Sie über das Kontextmenü oder das Menü *Ansicht/Darstellung/Konturen* zwischen farbiger oder Konturdarstellung umschalten.

3.5.4 Die Schriftart ändern

Die in Diagrammen verwendete Schriftart lässt sich über die Werkzeugleiste oder den Menüpunkt *Einstellungen/Schriftart…* verändern. Dabei sollte die Schriftgröße 10 bis 12 eingehalten werden.

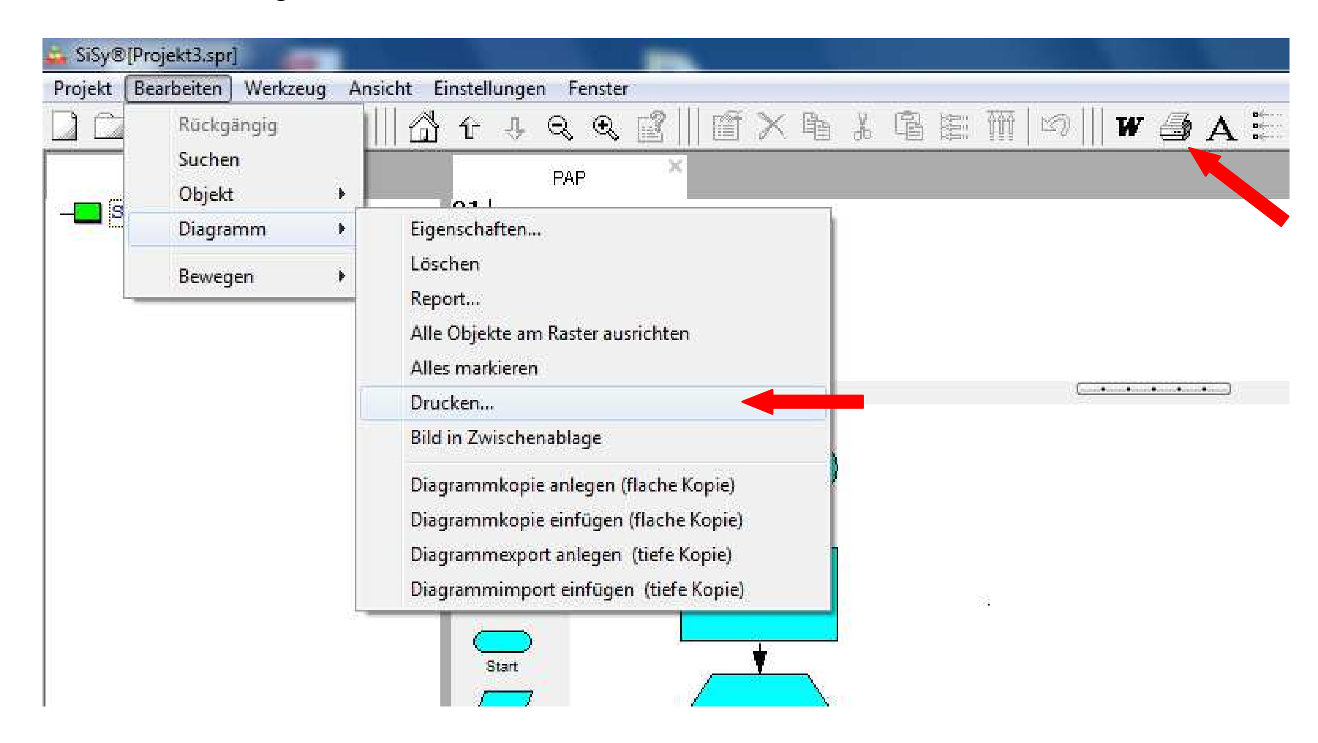
3.6 Druckfunktionen in SiSy

Sie haben in SiSy verschieden Möglichkeiten Projektinhalte wie Grafiken, Übersichten, Quellcodes oder ganze Projektdokumentationen zu drucken. Dabei ist zu berücksichtigen, dass in SiSy bestimmte Informationen wie zum Beispiel die Darstellung eines Programmablaufplanes sichtbare Elemente eines Diagramms sind und andere Teile wie zum Beispiel der Quellcode eines Elementes nur über Dialoge oder bei Selektierung des Elementes sichtbar sind. Je nachdem, welchen Inhalt Sie dokumentieren wollen, richtet sich die Auswahl der betreffenden Druckfunktion.

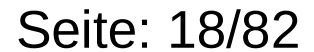
3.6.1 Diagramme drucken

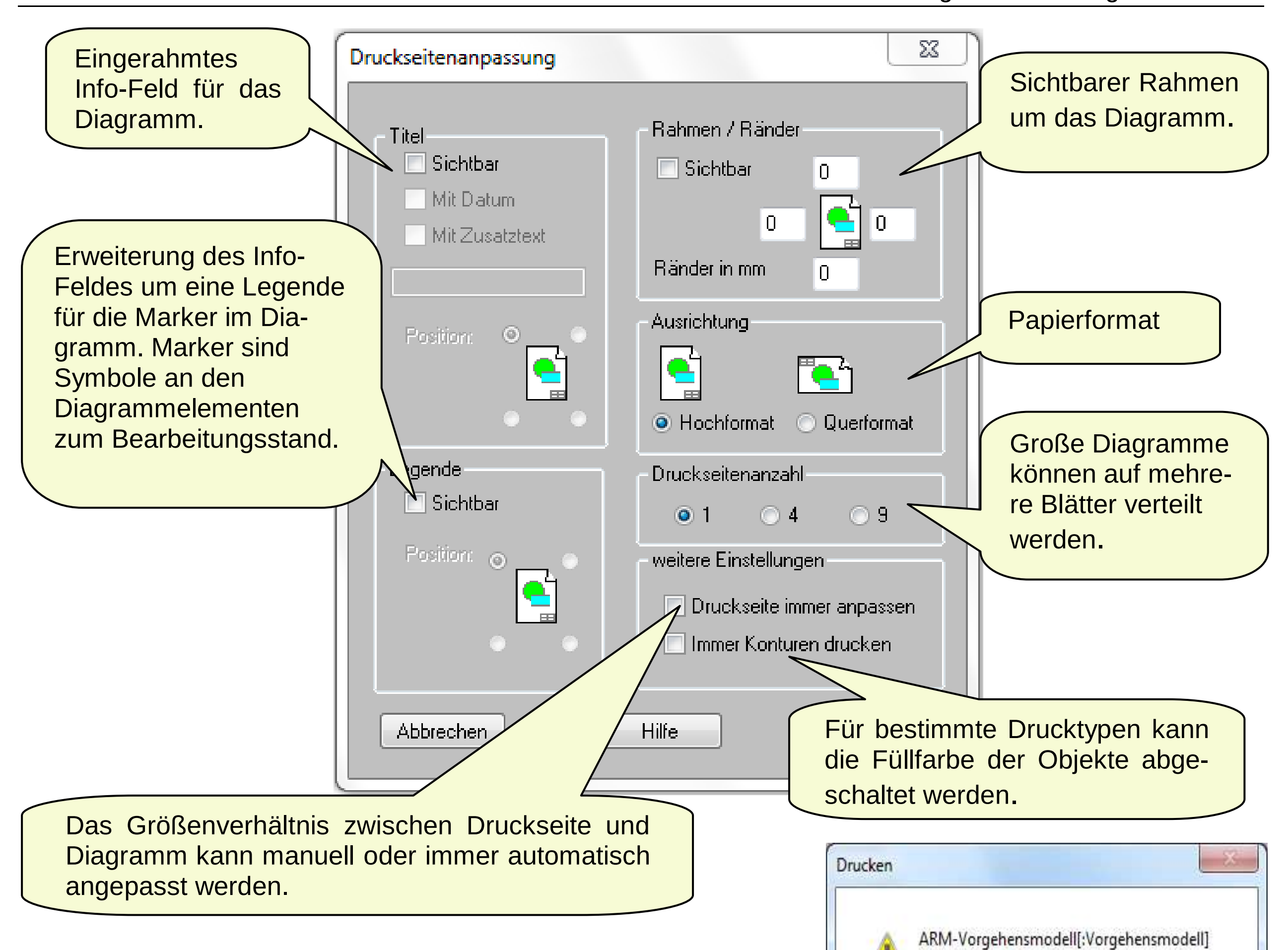
Wenn Sie ein einzelnes Diagramm, also den sichtbaren Inhalt eines Diagrammfensters drucken wollen, gehen Sie wie folgt vor:

- ggf. Projekt öffnen
- das gewünschte Diagramm öffnen
- die Menüfolge *Bearbeiten/Diagramm/Drucken…* wählen oder das Druckersymbol in der Werkzeugleiste aktivieren



Sie erhalten den Dialog zum Einrichten der Druckseite, wählen Sie die gewünschten Optionen aus!



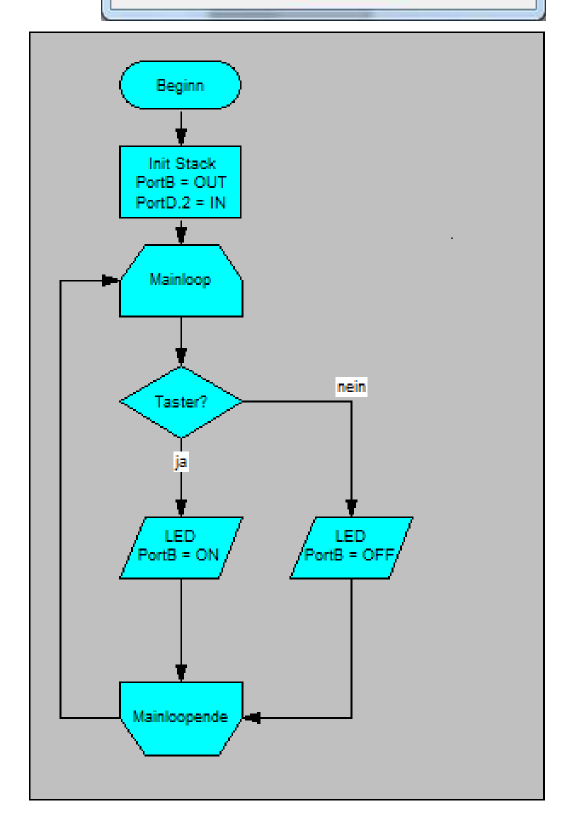


- Zum Drucken wählen Sie die Schaltfläche „OK". Der Ausdruck erfolgt auf dem Standarddrucker des Systems

- Im Hintergrund ist die Druckvorschau zu sehen. Der Druckvorgang kann hier abgebrochen werden, um die Einstellungen zu überarbeiten. Dabei kann die relative Position und das Größenverhältnis der Druckseite zum Diagramm verändert werden. Der Dialog zum Verändern der Einstellungen lässt sich per Doppelklick auf den Selektierungsmarken der Druckseite öffnen.

- Die Druckseitenansicht lässt sich über die Menüfolge *Ansicht/Druckseite* ein- und ausblenden

- Der Druckvorgang kann über das Druckersymbol in der Werkzeugleiste jederzeit gestartet werden



3.6.2 Grafiken und Inhalte drucken (QuickDok)

Viele Projektinformationen sind kein sichtbarer Teil von erstellten Diagrammen. Diese wurden über Dialoge und Masken eingegeben und stehen als Attribute in der Projektdatenbank zur Verfügung. Um diese Informationen auszudrucken, stellt SiSy für jeden Diagrammtyp eine Reportfunktion zur Verfügung, mit der die wesentlichen Informationen, Inhalte und Attribute des Diagramms und der Objekte in einem Diagramm, als Word-Dokument generiert werden.

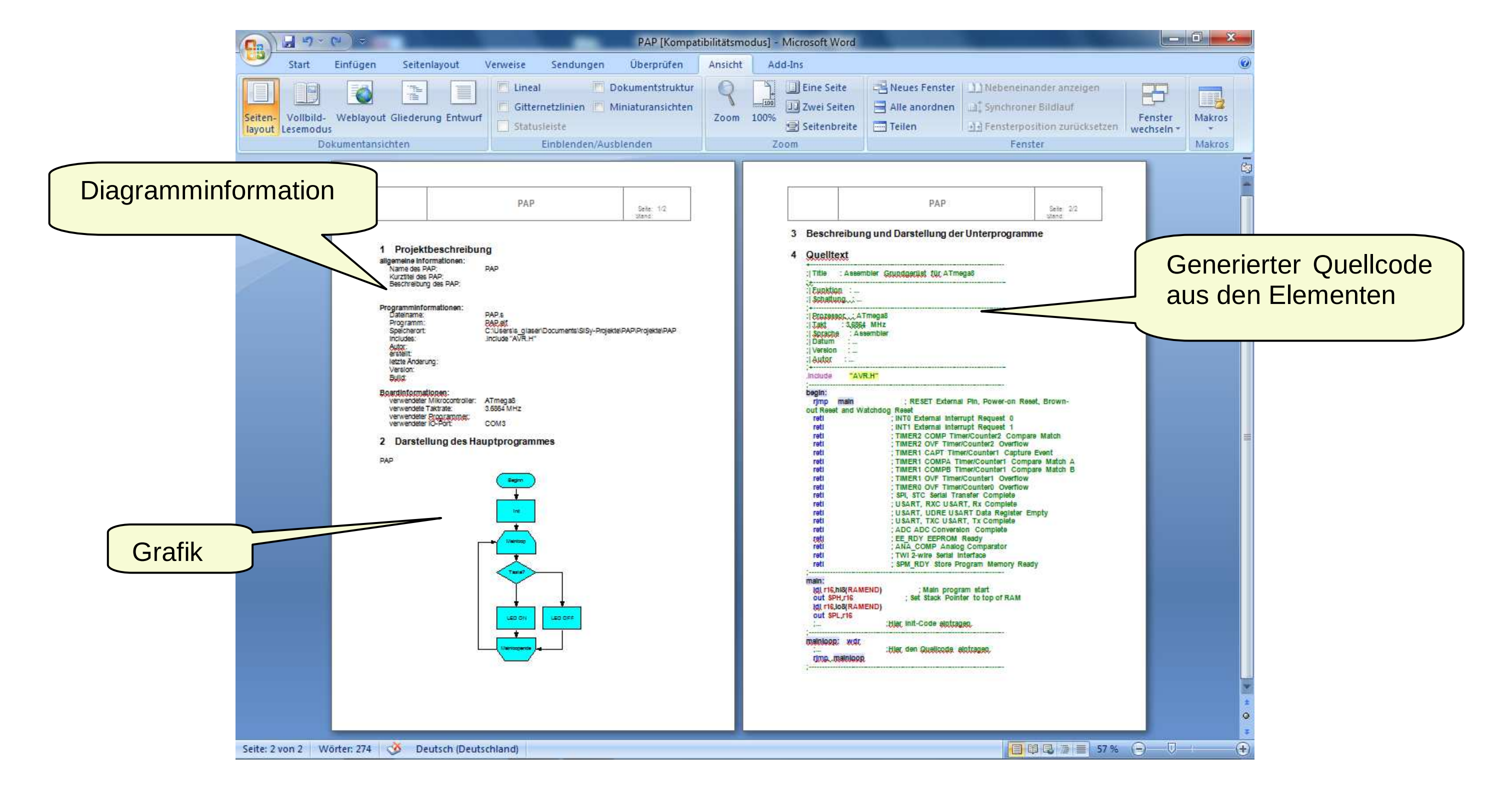
Für das Generieren des Word-Dokumentes muss auf dem PC MS Word installiert sein. Unterstützt wird die Version Word 2007.
Die Dokumentengenerierung basiert auf Makros.
Vorgehensweise in Word:

- Wählen Sie die Schaltfläche „Office“
- Menüfolge: *Word-Optionen / Vertrauensstellungscenter / Einstellungen für das Vertrauensstellungscenter / Einstellungen für Makros*
- Aktivieren Sie unter „Einstellungen für Makros“ den Auswahlpunkt „Alle Makros aktivieren“
- Setzen Sie im Unterpunkt „Entwicklermakroeinstellungen“ den Haken bei „Zugriff auf das VBA-Projektobjektmodell vertrauen“

Die Reportfunktion zum Generieren des Word-Dokumentes aktivieren Sie über das Symbol „QuickDok“ in der Werkzeugleiste.

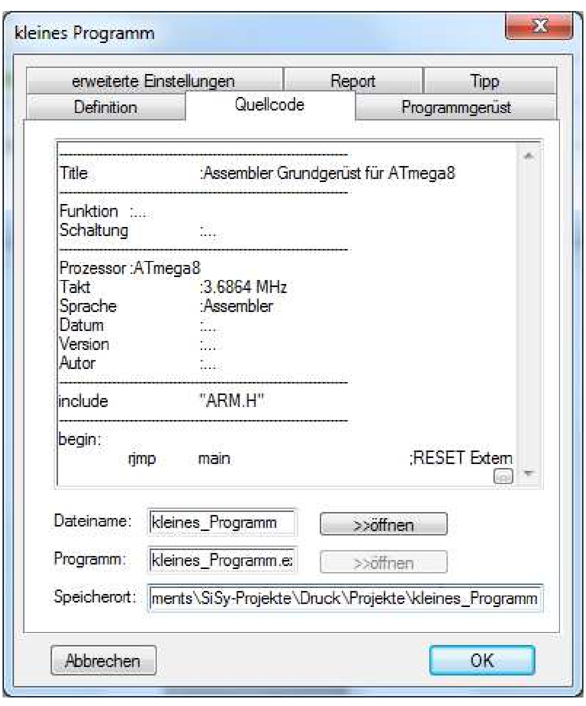
- Symbol (QuickDok) in der Werkzeugleiste wählen
- Aus der Liste der angebotenen Reportfunktionen auswählen
- Dateinamen für den Report vergeben
- Makros aktivieren





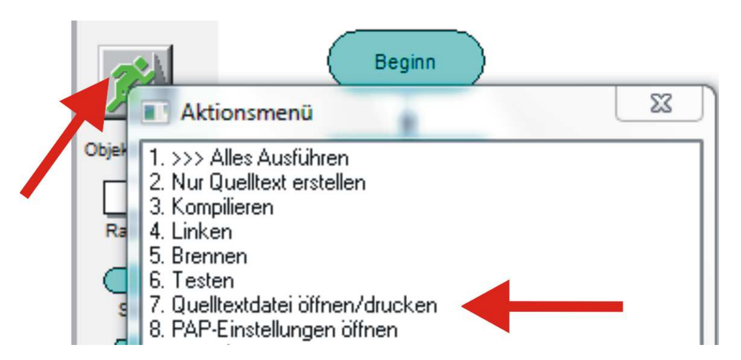
3.6.3 Nur Quellcodes drucken

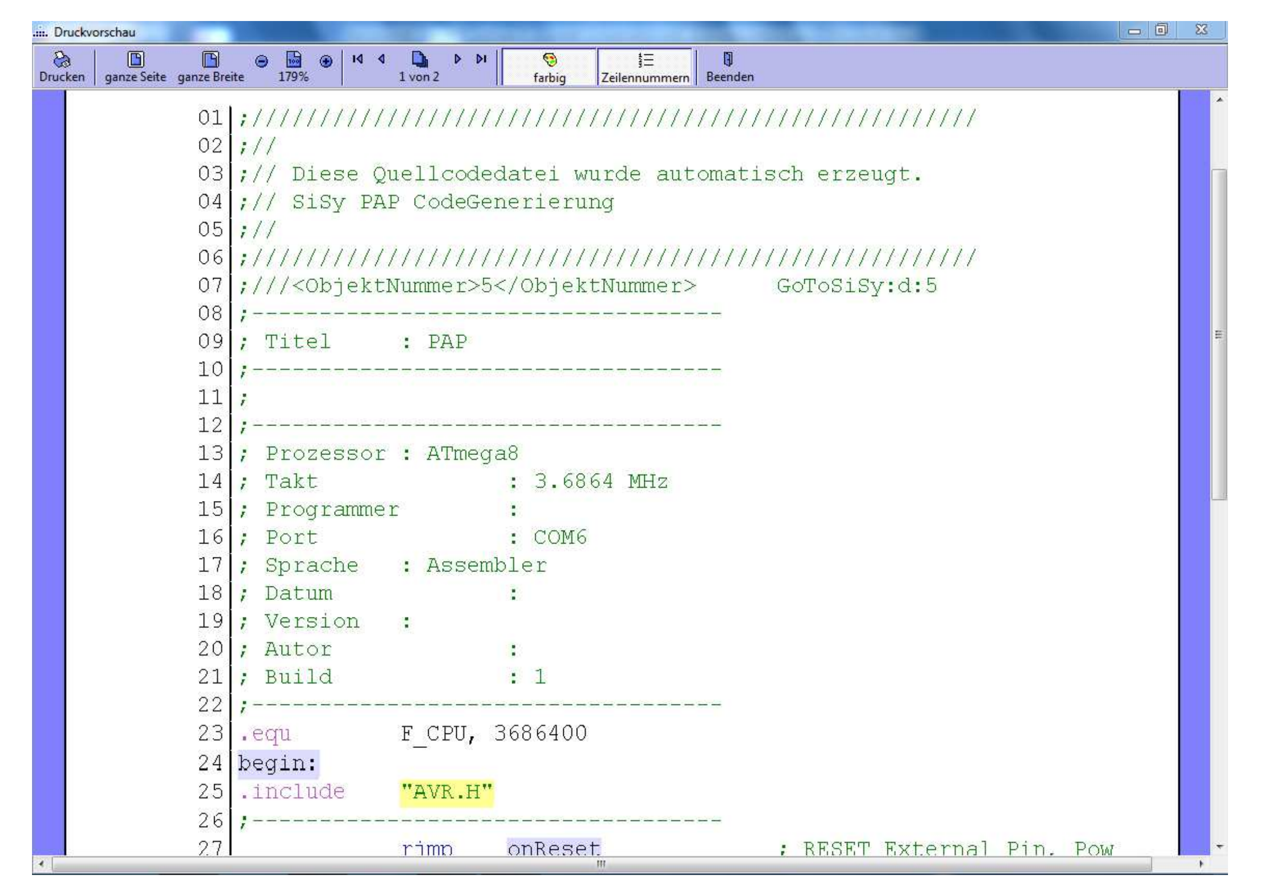
Für das Ausdrucken von Quellcodes bietet SiSy einen speziellen Quellcode-Druckassistenten. Damit ist es möglich Quellcodes formatiert, in Syntaxfarben und mit Zeilennummern auszudrucken.



- kleines Programm, Quellcode drucken
    - kleines Programm selektieren
    - rechte Maustaste „Definieren…“
    - Dialogseite „Quellcode“
    - Schaltfläche „>>öffnen“

- Programmablaufplan, Quellcode drucken
    - PAP öffnen
    - Aktionsmenü aktivieren
    - „Quellcodedatei öffnen/drucken“

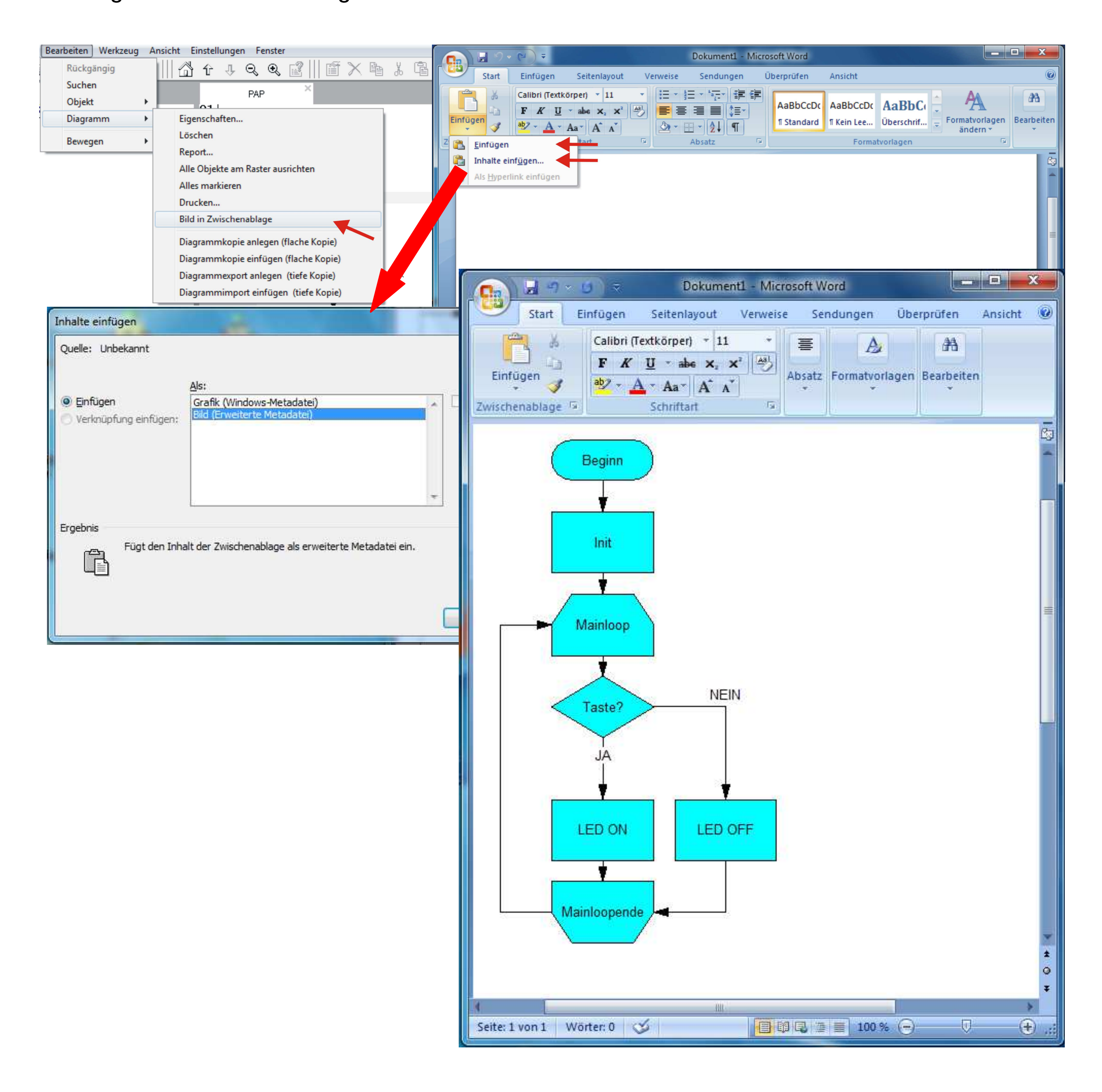




3.6.4 Nutzen der Zwischenablage

Oft ist es erforderlich in Projektdokumentationen die Diagramme als Bilder einzufügen. In SiSy werden die Diagramme nicht als Bilder gespeichert sondern zur Laufzeit aus den Modellinformationen generiert. Um die Bilder der Diagramme weiter zu verwenden, steht dem Anwender die Funktion „Bild in Zwischenablage“ zur Verfügung. Dabei erstellt SiSy eine skalierbare Vektorgrafik (WMF) und legt diese in die Zwischenablage (Copy). Die Grafik kann nun von anderen Anwendungen über den Befehl „Einfügen“ (Paste) beliebig weiter verwendet werden.

- gewünschtes Diagramm öffnen
- Menüfolge *Bearbeiten/Diagramm/Bild in Zwischenablage* wählen
- Zielanwendung, zum Beispiel Word öffnen
- Menüfolge *Start/Einfügen* oder *Start/Einfügen/Inhalte einfügen* wählen
- Gegebenenfalls einzufügendes Grafikformat wählen



3.7 SiSy Add-Ons verwalten

3.7.1 Einführung

Das Modellierungswerkzeug SiSy besteht aus seinen Kernkomponenten:

- Anwendungssystem SiSy mit
  - Rahmenanwendung
  - Assistent
  - Diagrammfenster
- Laufzeitbibliotheken
  - Datenbanktreiber (Repository)
  - Grafikbibliotheken
  - Metamodell-Engine
  - SiSy BASIC Interpreter
- Allgemeine Hilfe (dieses Benutzerhandbuch)

… und installierten Add-Ons.

Add-Ons enthalten zusätzliche Komponenten, SiSy-BASIC-Skripte, erweiterte Hilfen und spezielle Informationen für die Modellierung (Metamodell). Dabei können Add-Ons wiederum aus Add-Ons thematisch zusammengestellt werden. Diese zusätzlichen Bausteine werden während der Laufzeit vor allem durch die Metamodell-Engine verarbeitet. Dadurch können beliebige Modellierungsaufgaben durch SiSy verarbeitet werden. Es ist nur das entsprechende Add-On erforderlich. Für die aktuelle Version von SiSy sind zum Beispiel folgende Add-On Zusammenstellungen verfügbar:

- **AVR**, Programmierung von AVR Mikrocontrollern
  - einfache Programmierung mit dem Zeileneditor in Assembler und C
  - grafische Programmierung mit dem Programmablaufplan in Assembler, dem Struktogramm in C, dem UML-Klassendiagramm in C++ und dem UML-Statechart in C++
  - Codegenerierung mit dem myAVR Code-Wizard für Assembler und C
  - myAVR Werkzeuge für Test und Kommunikation mit AVR Controllern
- **SPRG**, einfache Programmierung für Konsolen- und GUI-Anwendungen
  - C / C++, JAVA, C#, PASCAL, Assembler
  - Integrierter Debugger
- **UML**, objektorientierte Systementwicklung für Konsolen- und GUI-Anwendungen
  - Typische UML-Diagramme z.B.: Klassendiagramm, Anwendungsfalldiagramm, Aktivitätsdiagramm, Zustandsdiagramm, Sequenzdiagramm, u.a.
  - Codegenerierung für C++, C#, JAVA
  - Integrierter Debugger mit Modellnachführung im Schrittbetrieb
- **SVL**, Smart Visual Library
  - Einfache bausteinartige Entwicklung von Windows-Anwendungen (RAD Rapid Application Development) im UML Klassendiagramm
- **SysML**, System Modeling Language
  - Fachübergreifende Systementwicklung mit dem OMG Sprachstandard
- **SA/SD**, Strukturierte Techniken zur Systementwicklung
  - Round Trip Engineering mit den klassischen Analyse und Entwurfswerkzeugen bis zur Codegenerierung, Strukturierte Analyse, Entity Relationship Diagramm mit SQL-Generierung und Struktogramme mit Codegenerierung
- **DOKGEN**, Dokumentengenerierung
  - Generieren von Dokumentationen direkt aus den Modellen heraus.
- **GPM / QM**, Geschäftsprozessmodellierung und Qualitätsmanagement
  - Modellierung von integrierten Managementsystemen
- **ARM,** Programmierung von ARM Mikrocontrollern
  - einfache Programmierung mit dem Zeileneditor in C
  - grafische Programmierung mit dem Struktogramm in C, dem UML-Klassendiagramm in C++ und dem UML-Statechart in C++
  - Integrierter Debugger für ARM Mikrocontroller

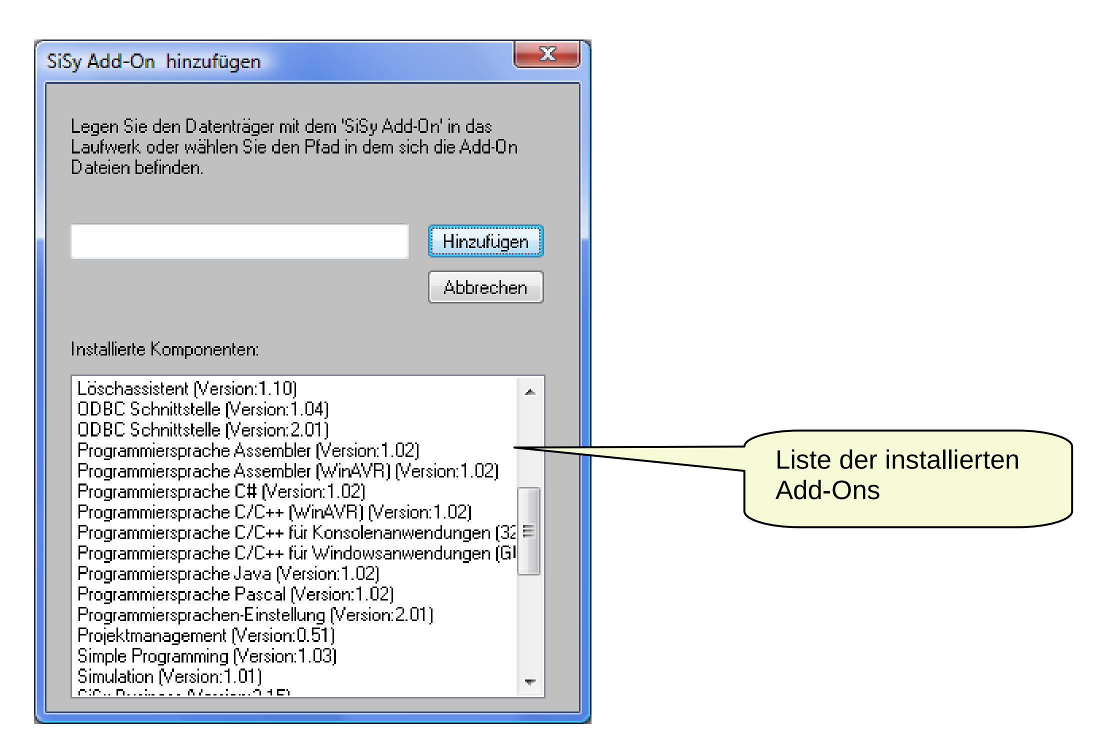
Aus den verfügbaren Add-Ons werden entsprechend der unterschiedlichen Einsatzgebiete des Modellierungswerkzeuges SiSy spezielle Ausgaben zusammengestellt. Die jeweiligen Ausgaben können jederzeit durch weitere Add-Ons ergänzt werden. Es sind unter anderen folgende SiSy-Ausgaben verfügbar (Stand August 2012):

Ausgabe

- SiSy Professional
  umfasst alle verfügbaren Add-Ons
- SiSy Business
  umfasst Add-Ons für Prozessmodellierung, Qualitäts- und Projektmanagement
- SiSy Developer
  umfasst umfangreiche Add-Ons zur Systementwicklung
- SiSy AVR ++
  umfasst die Add-Ons AVR, ausgewählte Teile der UML, SVL und SysML
- SiSy AVR
  beinhaltet nur das Add-On AVR
- SiSy ARM
  umfasst die Add-Ons ARM, ausgewählte Teile der UML und SysML

3.7.2 Add-Ons anzeigen

Um die Liste der installierten Add-Ons anzuzeigen nutzen Sie den Menüpunkt *Einstellungen/Add-On hinzufügen*. Um den Dialog zu verlassen wählen Sie die Schaltfläche „Abbrechen".

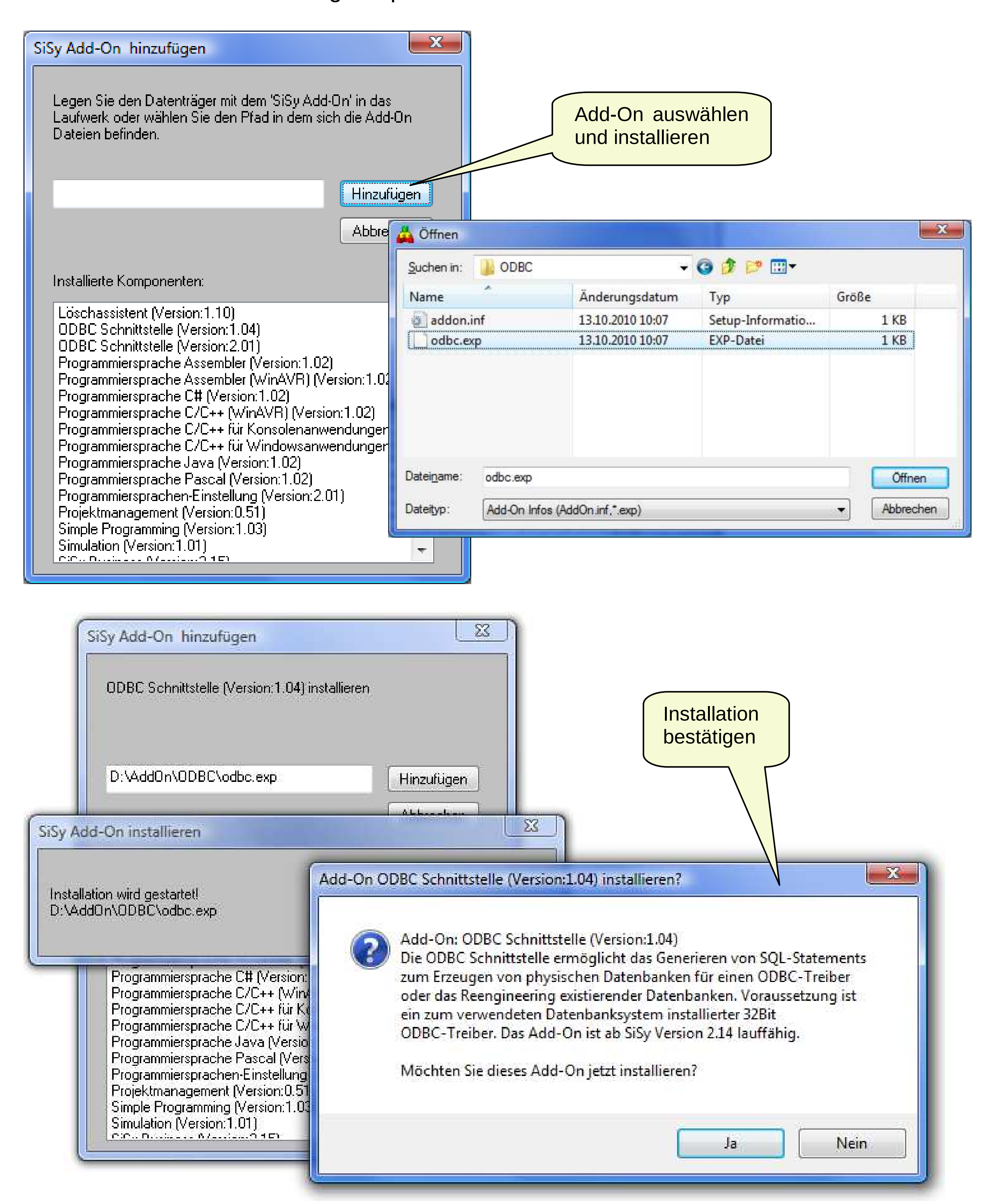


Hinweis:

Beachten Sie, dass Änderungen an den Add-Ons in jedem Fall Auswirkungen auf die Funktionalität von SiSy und abhängiger Add-Ons haben.

3.7.3 Add-Ons hinzufügen

Für das Hinzufügen von Add-Ons nutzen Sie den Menüpunkt *Einstellungen/Add-On hinzufügen*. Wählen Sie im Add-On Dialog die Schaltfläche „Hinzufügen“. SiSy Add-Ons besitzen die Dateierweiterung *.exp.



Hinweis:

Für die Installation von Add-Ons sollten Sie Administratorrechte besitzen. Beachten Sie, dass Änderungen an den Add-Ons in jedem Fall Auswirkungen auf die Funktionalität von SiSy und abhängiger Add-Ons haben.

3.8 Zusätzliche Werkzeuge

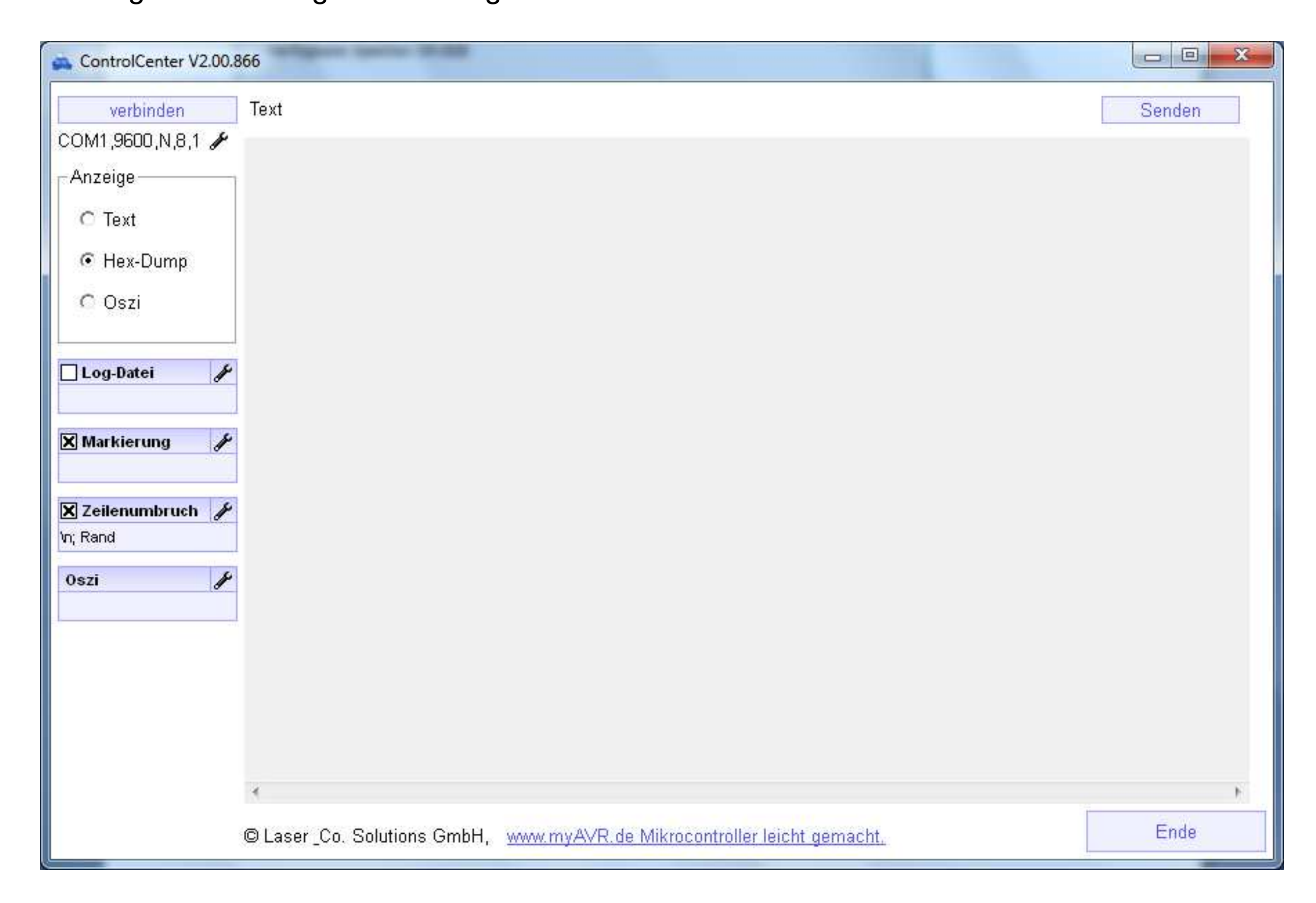
3.8.1 Einführung

Bestimmte Add-Ons installieren als zusätzliche Komponenten externe Programme. Diese sind nach der Installation über den Menüpunkt *Werkzeuge* erreichbar.

3.8.2 Werkzeuge für eingebettete Systeme

Das Mikrocontroller ControlCenter

Das Mikrocontroller Controlcenter ist ein universelles Terminalprogramm zur Kommunikation mit dem myAVR Board und anderen Mikrocontrollerapplikationen, die über eine serielle Schnittstelle (UART) oder USB Anbindung mit virtuellem COM-Port zum PC verfügen. Es kann für Test- und Debug-Meldungen sowie Visualisierung und Protokollierung von Messdaten genutzt werden. Dazu bietet das Mikrocontroller Controlcenter umfangreiche Konfigurationsmöglichkeiten.



Sie finden im Kapitel „Das Mikrocontroller Controlcenter“ ausführliche Informationen für dessen Handhabung. Das Controlcenter kann einzeln über das Menü *Werkzeuge* oder im Zusammenhang mit einer Mikrocontrolleranwendung aus dem jeweiligen Diagramm für die Entwicklung des Programms (z.B. Klassendiagramm, Programmablaufplan, usw.) aufgerufen werden.

Hinweis:

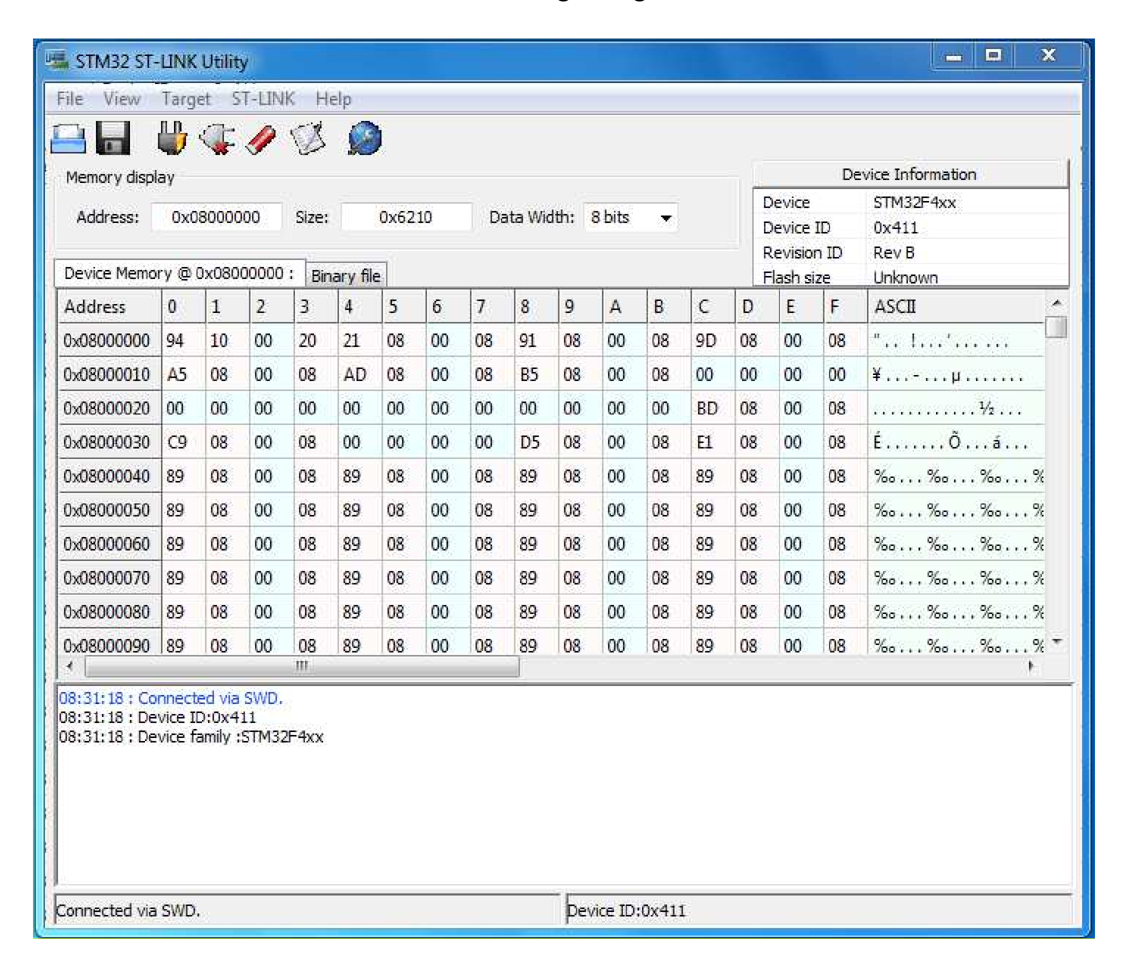
Das Mikrocontroller Controlcenter verfügt über spezielle Fähigkeiten die myAVR Programmer vom Typ MK2 und MK3 anzusteuern. Informieren Sie sich über die Möglichkeiten der Spannungsversorgung und der Kommunikation dieser Programmer in den jeweiligen technischen Beschreibungen. Diese Dokumente stehen Ihnen im Downloadbereich von www.myAVR.de zur Verfügung.

ST-Link

ST-Link ist ein Werkzeug zum Programmieren von ARM-Mikrocontrollern. Damit können Sie auch den Programmspeicher komfortabel programmieren.

Brennen:

Der Inhalt der ausgewählten Dateien und des Speichers des ARM-Mikrocontrollers wird im mittleren Bereich des Fensters angezeigt.



Ausgabe:

Die Ausgabe befindet sich im unteren Teil des ST-Link Fensters, die Aktionen werden im Ausgabefenster protokolliert. Es werden Informationen und Dauer der Aktion angezeigt. Zustand und Ergebnis werden durch Signalfarben und Ausgaben dargestellt.

Signalfarben:

rote Schrift: keine Verbindung
blaue Schrift: Zustandsinformation
schwarze Schrift: verbundener Controller
grüne Schrift: erfolgreiches Brennen

Speicher anzeigen und ändern:

Neben dem Geräte-Informationsbereich enthält das Hauptfenster zwei weitere Bereiche:

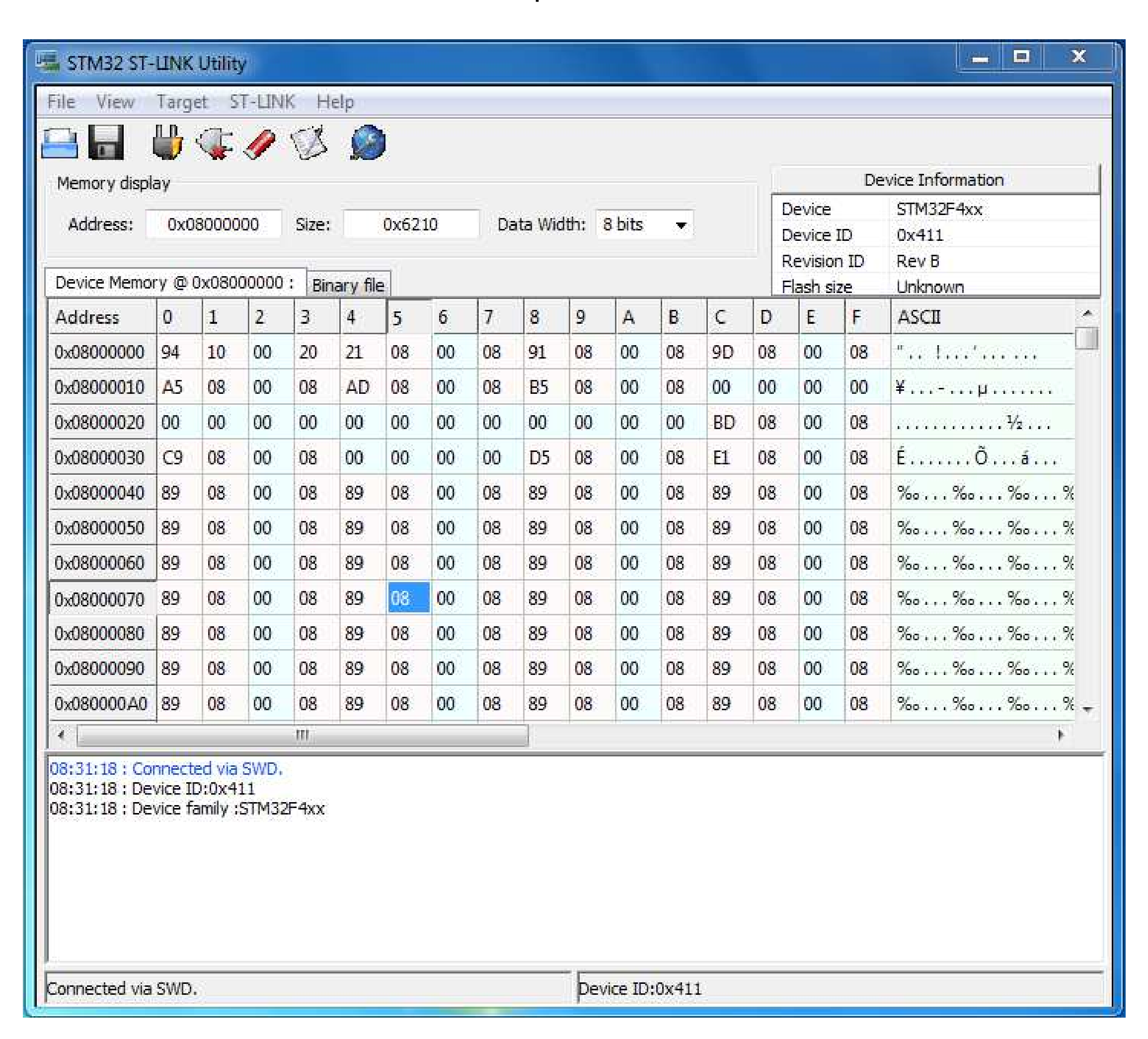
- Memory Display
- Memory Data

Memory Display:
Dieser Bereich enthält drei Eingabefelder:

- *Address*: Startadresse des Speichers, von der Sie lesen wollen.
- *Size*: Anzahl der zu lesenden Daten in Byte (Hexadezimalformat).
- *Data Width*: Breite der angezeigten Daten (8-bit, 16-bit oder 32-bit).

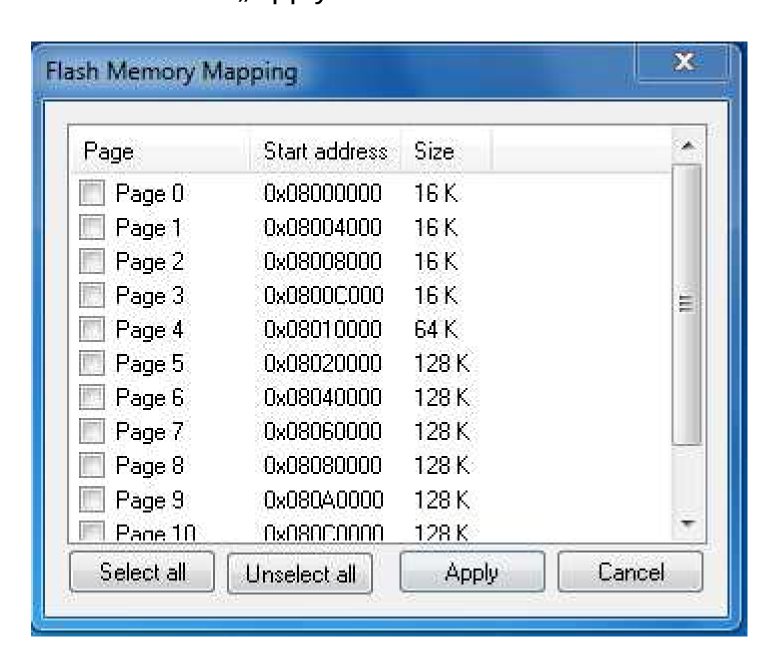
Memory Data:
In diesem Bereich werden die Daten aus einer binären Datei oder aus dem Speicher eines angeschlossenen Gerätes ausgelesen. Sie können den Inhalt der Datei vor dem Brennen ändern.

- Um den Inhalt der binären Datei anzeigen zu lassen, klicken Sie auf *File/Open file ...*
- Um den Speicherinhalt eines verbundenen Geräts auszulesen und anzeigen zu lassen, geben Sie die Startadresse, Datengröße und die Datenbreite in die Felder ein und bestätigen dann mit „Enter".
- Nach dem Lesen von Daten können Sie auch jeden Wert verändern, wenn Sie auf den Wert doppelklicken. Die betroffene Zelle wird markiert.
- Sie können auch den Speicherinhalt in eine binäre Datei über das Menü *File/ Save file as...* abspeichern.

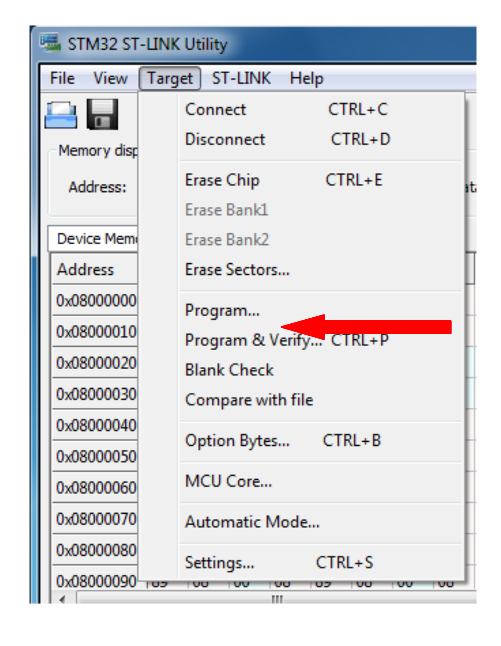


Flash-Speicher löschen:
Es gibt zwei Vorgehensweisen den Flash-Speicher zu löschen.

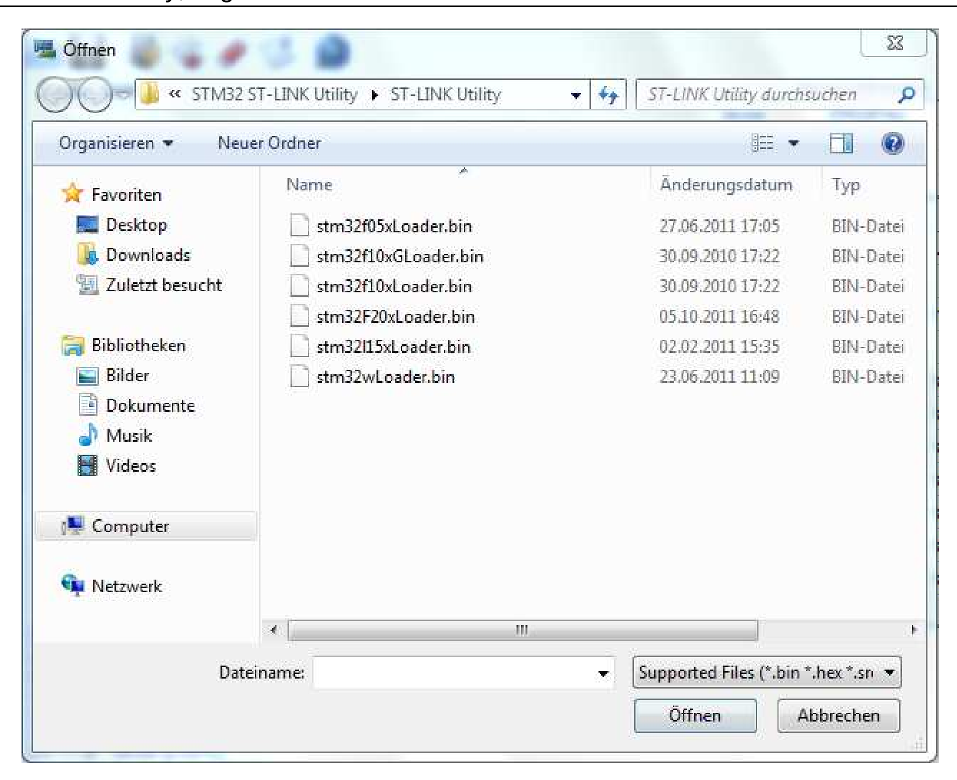
1. Gesamten Flash löschen:
   - Löschen Sie alle Flash-Speichersektoren des angeschlossenen Gerätes. Das wird über das Menü *Target/Erase Chip* ausgeführt.
2. Sektorweise den Flash löschen:
   - Um die Sektoren auswählen zu können klicken Sie im Menü *Target* auf *Erase Sectors….* Das "Flash Memory Mapping" Fenster zeigt die ausgewählten Sektoren des Flashs an.
     - „Select all" markiert alle Sektoren
     - „Deselect all" macht die Markierung rückgängig
     - „Cancel" bricht den Löschvorgang ab
     - „Apply" löscht alle markierten Sektoren



Programm brennen:
Das STM32 ST Link kann *.bin, *.hex, oder *.srec Dateien in den Flash oder RAM-Speicher brennen. Um das zu tun, führen Sie diese Schritte aus:

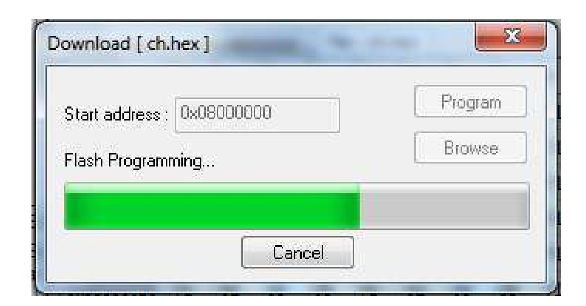


1. Klicken Sie auf *Target/Programm...* (oder *Target/Programm & verify…*, wenn nach dem Brennen die Daten überprüft werden sollen).
2. Nun öffnet sich das „Open file" Dialogfenster, in diesem können Sie die Datei auswählen und mit den „Open"-Button die Datei öffnen.



3. In dem „Device programming“ Dialogfenster, geben Sie die Startadresse ein, ab welcher das Programm gebrannt werden soll, es kann eine Flash oder RAM-Adresse sein

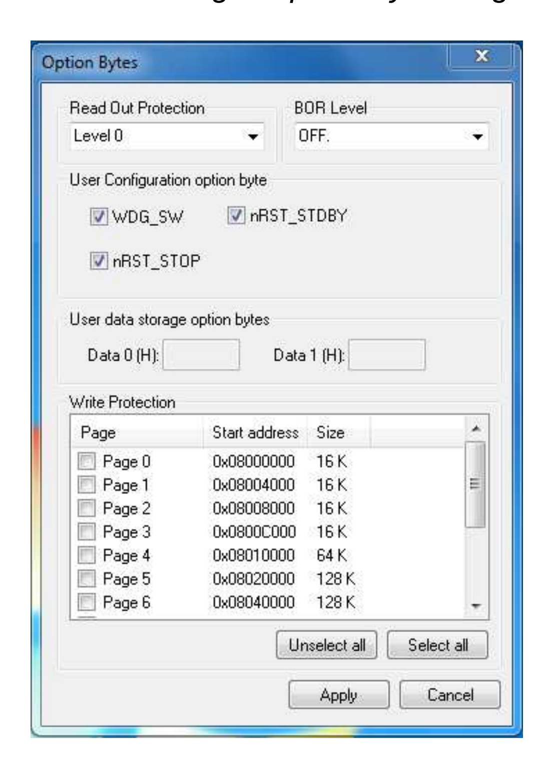
4. Zum Schluss klicken Sie auf den „Program“-Button und das Programm wird gebrannt. Wenn Sie *Target/Programm & verify…* ausgewählt wurde, wird das Programm nach dem Brennen überprüft.



Hinweise:

1. Die STM32F2 und STM32F4 Reihe haben unterschiedliche Programmierverfahren, die von der Spannungsversorgung der MCU abhängt. ST-Link verwendet die standardmäßige MCU-Spannungsversorgung. Unter dem Menü *Target/Settings…* kann die Spannungsversorgung geändert werden.
2. Wenn das Gerät schreibgeschützt gelesen wird, dann wird der Schreibschutz deaktiviert. Wenn einige Flash-Sektoren schreibgeschützt sind, dann wird der Schreibschutz während der Programmierung deaktiviert und nach der Programmierung wieder aktiviert.

Option-Bytes-Konfiguration:
Das STM32 ST-Link kann alle Option-Bytes über die „Option Bytes“ Dialogbox ändern, das über *Target/Option Bytes…* geöffnet werden kann.



Der „Option Bytes“ Dialogbox enthält die folgenden Einstellungsmöglichkeiten:

1. *„Read Out Protection“:* Verändert den Leseschutz des Flash-Speichers. Für STM32F2, STM32F4 und STM32L1 Geräte sind folgende Schutzniveaus verfügbar:
   - Level 0: kein Leseschutz
   - Level 1: Flash Leseschutz aktiviert
   - Level 2: Flash Leseschutz und alle Debug-Eigenschaften deaktiviert

   Für die anderen Geräte kann der Leseschutz nur aktiviert oder deaktiviert werden.
2. *„BOR Level“* (Brown Out Reset-Pegel): Diese Liste enthält unterschiedliche Spannungseinstellungen, beim Unterschreiten der Spannungsgrenze wird das Gerät in den Reset-Modus gesetzt.
   Diese Auswahl ist nur für das angeschlossene STM32 L1 Geräte gültig.
   Für Geräte mit niedriger Spannungsversorgung kann aus 5 VBOR Einstellungen ausgewählt werden:
   - BOR LEVEL 1:
     Reset-Schwelle im Spannungsbereich von 1,69 bis 1,8 V
   - BOR LEVEL 2:
     Reset-Schwelle im Spannungsbereich von 1,94 bis 2,1 V
   - BOR LEVEL 3:
     Reset-Schwelle im Spannungsbereich von 2,3 bis 2,49 V
   - BOR LEVEL 4:
     Reset-Schwelle im Spannungsbereich von 2,54 bis 2,74 V
   - BOR LEVEL 5:
     Reset-Schwelle im Spannungsbereich von 2,77 bis 3,0 V

Für STM32F2 und STM32F4 Geräte können 4 programmierbare VBOR Einstellungen ausgewählt werden:

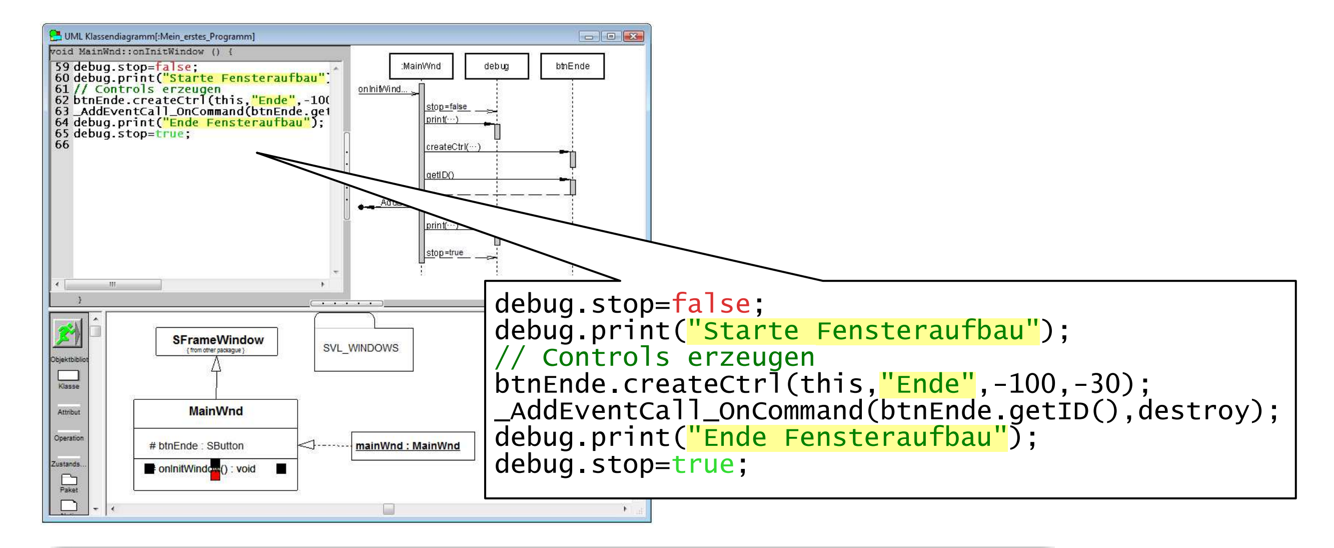
- BOR LEVEL 3:
  Spannungsversorgung im Bereich von 2,70 bis 3,60 V
- BOR LEVEL 2:
  Spannungsversorgung im Bereich von 2,40 bis 2,70 V
- BOR LEVEL 1:
  Spannungsversorgung im Bereich von 2,10 bis 2,40 V
- BOR off:
  Spannungsversorgung im Bereich von 1,62 bis 2,10 V

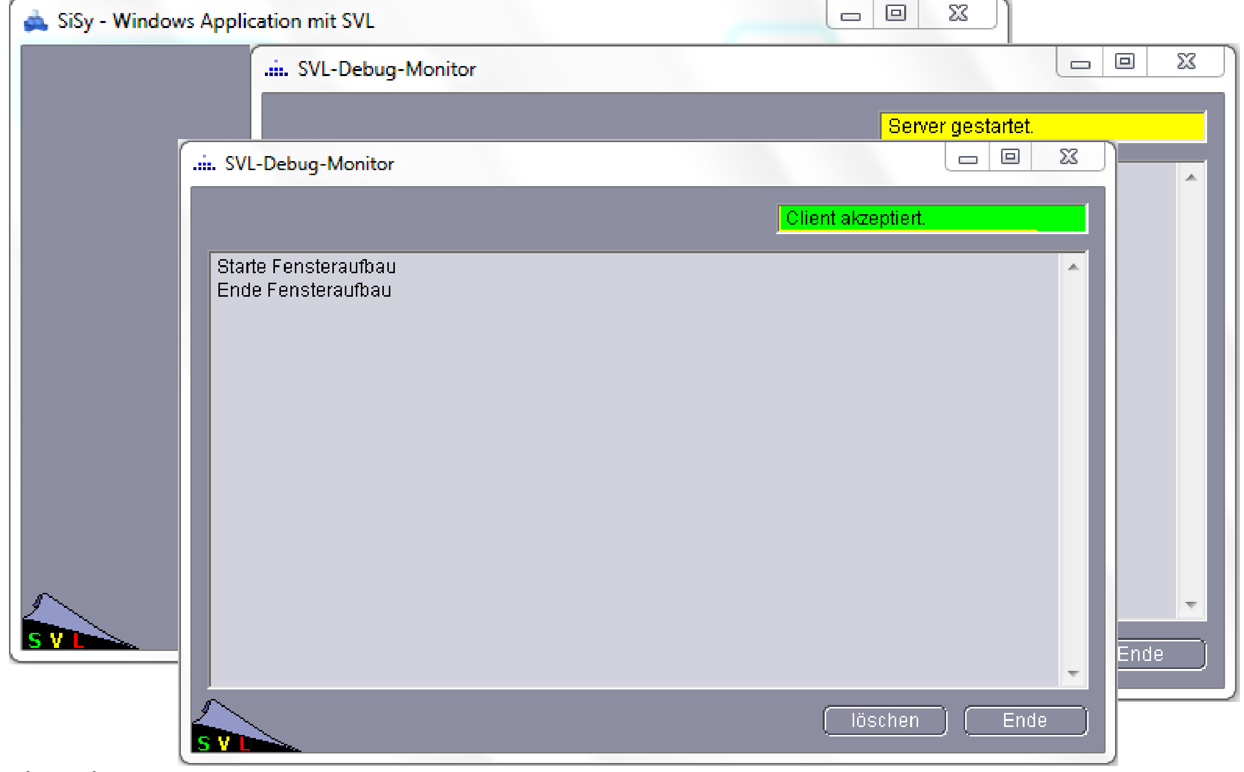
3. *„User Configuration option byte“:*
   - WDG_SW: Wenn aktiviert, muss der Watchdog durch die Software aktiviert werden, sonst wird er automatisch beim Einschalten aktiviert.
   - nRST_STOP: Wenn nicht aktiviert, wird der Reset im Standby-Modus generiert. Wenn aktiviert, wird kein Reset generiert beim Eintreten in den Standby-Modus.
   - nRST_STDBY: Wenn nicht aktiviert, wird ein Reset erzeugt beim Betreten des Stop-Modus (alle Uhren sind gestoppt). Wenn aktiviert, wird kein Reset generiert beim Betreten des Stop-Modus.
   - BFB2: Wenn nicht aktiviert und die Boot-Pins gesetzt sind, um das Gerät beim Start vom Anwender-Flash zu booten, startet das Gerät von der „Flash bank“ 2, sonst von der „Flash bank“ 1. Diese Option ist nur aktiviert, wenn das Gerät zwei Flash-Banks besitzt.
4. „*User data storage option bytes*“: Enthält 2 Byte für den “User storage”. Diese Bytes sind nicht verfügbar auf STM32F2, STM32F4 und STM32L1 Geräten
5. „*Write Protection*“: Die Flash-Sektoren sind geräteabhängig gruppiert. Hier kann der Schreibschutz für die einzelnen Sektoren aktiviert werden.

3.8.3 SVL Werkzeuge

Der SVL-Debug-Monitor

Für die Fehlersuche im laufenden Betrieb einer Windowsanwendung die auf der SVL-Klassenbibliothek basiert, steht Ihnen der SVL-Debug-Monitor zur Verfügung. In der folgenden Darstellung sehen Sie eine typische Anwendung des Debugmonitors. Dabei fungiert der Debugmonitor als lokaler Server und die Anwendung als Client. Der Debugmonitor muss immer zuerst gestartet werden und danach erst die Anwendung. Sie können in der SVL Hilfe unter den Stichwort „SDebug“ detaillierte Informationen zur Nutzung der Debugschnittstelle abrufen.



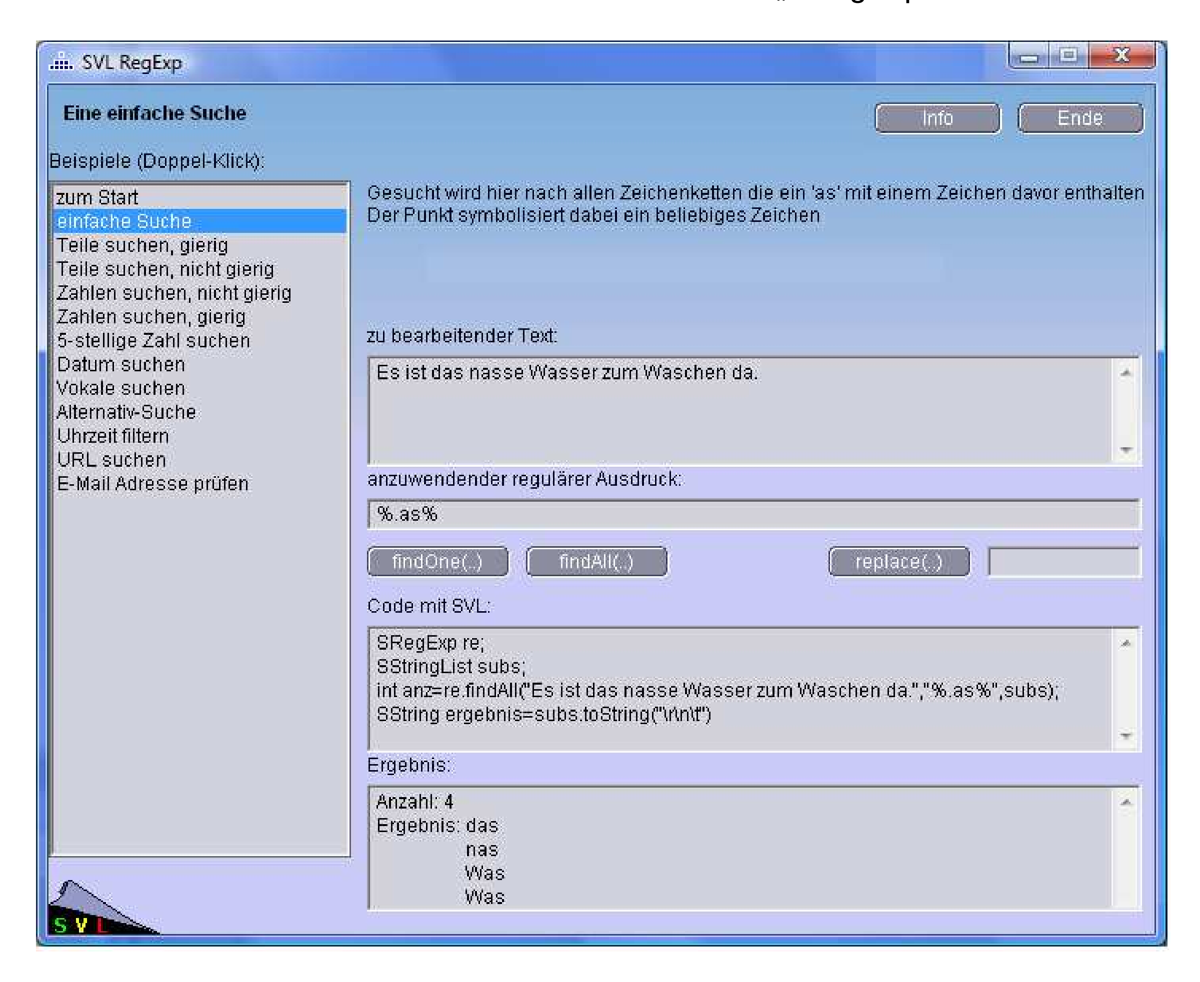


Hinweis:

SVL Anwendungen können nicht mit jeder SiSy Ausgabe erstellt werden. Sie benötigen dazu die entsprechenden Add-Ons beziehungsweise die entsprechende Ausgabe von SiSy. Bei der Nutzung des Debugmonitors kann es zu Warnhinweisen Ihrer Firewall kommen. Es ist für die korrekte Funktion des Debugmonitors notwendig, die Verbindung zwischen Anwendung und Debugmonitor nicht zu blockieren.

Das SVL-Werkzeug RegExp

Reguläre Ausdrücke sind Regeln für die Suche in Zeichenketten. Man definiert also mit dieser Syntax Suchmuster. Benutzen Sie dieses Werkzeug um die Syntax eines gewünschten regulären Ausdruckes zu testen. Sie erhalten zusätzlich den C++ Code für die Anwendung des getesteten regulären Ausdrucks mit der SVL. Weitere Informationen finden Sie in der SVL-Hilfe unter den Stichwort „SRegExp“.



Hinweis:

SVL Anwendungen können nicht mit jeder SiSy Ausgabe erstellt werden. Sie benötigen dazu die entsprechenden Add-Ons beziehungsweise die entsprechende Ausgabe von SiSy.

3.8.4 Weitere Werkzeuge

Entsprechend der installierten Add-Ons Ihrer SiSy Ausgabe können weitere Werkzeuge verfügbar sein.

- mySmartUSB Terminal
- mySmartUSB-Light SupportBox
- myAVR BitmapEditor
- UML Tools
- Language Tools
- Export/Import Tools
- …

4 Die Hilfefunktionen in SiSy

Nutzen Sie die zahlreichen Hilfen und Vorlagen, die SiSy dem Entwickler bietet!

4.1 Der Assistent

Der Assistent ist hilfreich bei der Unterstützung und Führung des Nutzers im Umgang mit SiSy. Er befindet sich standardmäßig im linken, unteren Bildschirmbereich. Der Assistent kann über die Werkzeugleiste (-Symbol) geöffnet werden, falls dies nicht beim Programmstart erfolgte. Dieser Assistent begleitet Sie im gesamten Projekt. Sie erhalten immer passende Informationen zum aktuellen Diagrammtyp und haben die Möglichkeit, durch verschiedene Links weitere Hilfethemen aufzurufen oder Vorlagen in Ihr Projekt zu laden.

Beachte: Der Assistent ist auf die jeweilige Ausgabe von SiSy, die verfügbaren Add-Ons und das gewählte Modell bezogen.

Bedeutung der verwendeten Symboliken im Assistenten:

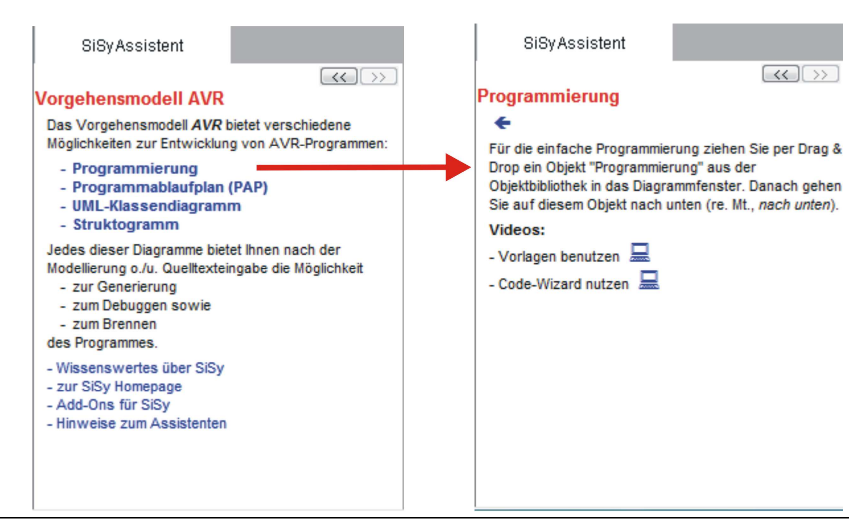
	weitere Informationen anzeigen; öffnet eine Hilfedatei (*.chm, *.hlp, *htm)
	Demovideo zur Handhabung zeigen; öffnet eine Animation oder Videomitschnitt der Bildschirmarbeit (AVI, ScreenCam- oder FLASH-Film)
	entsprechendes Diagramm öffnen, das so geöffnete Diagramm kann über die Schließen-Schaltfläche des Diagramms wieder geschlossen werden;
Beispiel 1	eine Diagrammvorlage laden;
	Vorschau zur Diagrammvorlage;
	zurück zur Startseite des Assistenten;
	ein kleines Skript zu Arbeitsschritten anzeigen;
	nächster Schritt (Arbeitsschritt);
	Verweis auf den Standardassistenten dieser Sicht;
blaue Schrift	und
blaue Symbole	sind mit Links hinterlegt und können per Mausklick aktiviert werden.

SiSyAssistent

Hinweise zum Assistenten

Bedeutung der verwendeten Symboliken im Assistenten:
weitere Informationen;
Demo zeigen;
entsprechendes Diagramm öffnen, das so geöffnete Diagramm über den Schließen-Button des Diagramms wieder schließen;
Vorschau zur Vorlage;
Vorlage Vorlage laden;
zurück zur Startseite;
zu Arbeitsschritten;
nächster Schritt;
Verweis auf den Standardassistenten dieser Sicht;
blaue Schrift und Symbole sind mit Links hinterlegt.

Beispiele für Assistenten:

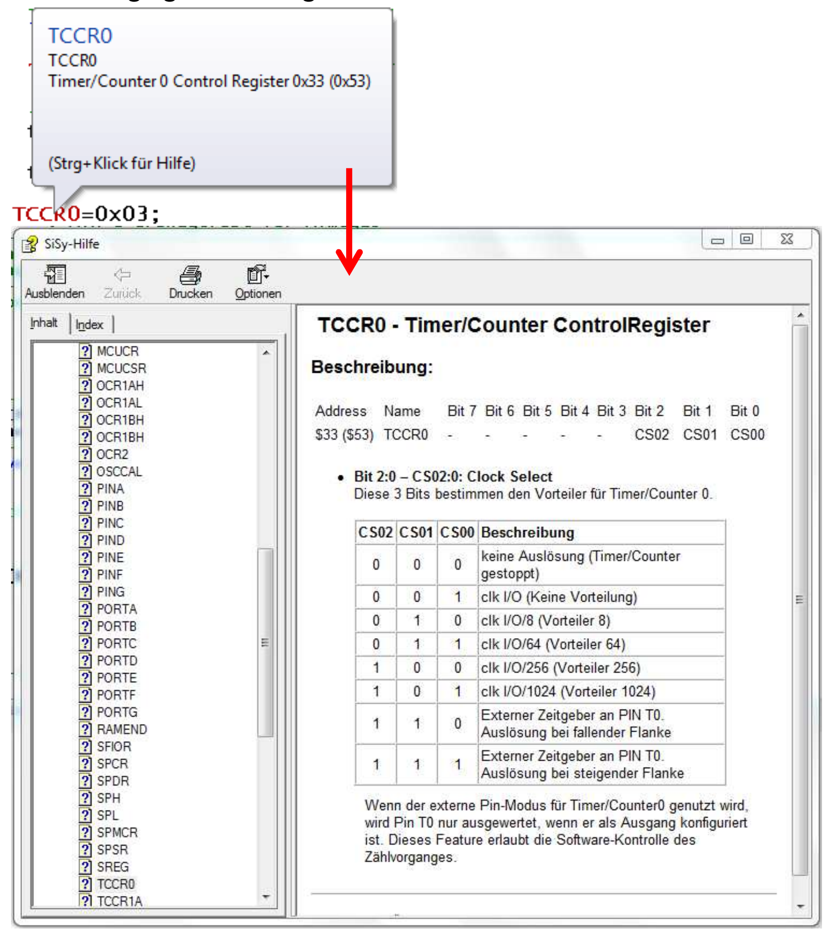


4.2 Die Online-Hilfe

Bei der Eingabe von Quellcode im dafür vorgesehenen Editorfenster werden reservierte Worte (Bezeichner, Schlüsselworte) der gewählten Programmiersprache durch verschiedenfarbiges Einfärben (Syntaxfarben) hervorgehoben. Zu den hervorgehobenen Bezeichnern existiert in der Regel auch eine kurze Online-Hilfe und eine ausführlichere Hilfe. Die Online-Hilfe ist ein Pop-Up-Fenster, welches automatisch eingeblendet wird, wenn Sie mit dem Mauszeiger auf einen Befehl zeigen. In dem Pop-Up ist eine kurze Hilfestellung zu dem Bezeichner eingeblendet. Steht eine Hilfe mit ausführlicheren Informationen zur Verfügung, wird diese in dem Pop-Up (STRG+Klick für Hilfe) angezeigt.

Schlüsselwort-Hilfe

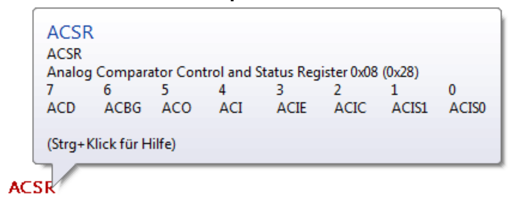
Bei der Eingabe von bekannten Registernamen wird die Bezeichnung des Registers und dessen Adresse eingeblendet. Durch betätigen der Taste „STRG" und gleichzeitigem Klick auf das Register öffnet eine Hilfedatei mit dem entsprechenden Hilfethema zu dem eingegeben Register.



CS02	CS01	CS00	Beschreibung
0	0	0	keine Auslösung (Timer/Counter gestoppt)
0	0	1	clk I/O (Keine Vorteilung)
0	1	0	clk I/O/8 (Vorteiler 8)
0	1	1	clk I/O/64 (Vorteiler 64)
1	0	0	clk I/O/256 (Vorteiler 256)
1	0	1	clk I/O/1024 (Vorteiler 1024)
1	1	0	Externer Zeitgeber an PIN T0. Auslösung bei fallender Flanke
1	1	1	Externer Zeitgeber an PIN T0. Auslösung bei steigender Flanke

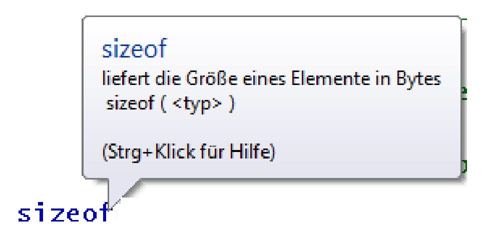
Bit-Hilfe

Wenn für ein Register detaillierte Informationen zu der Bedeutung/Funktion der einzelnen Bits vorliegen, wird zusätzlich im Pop-Up eine Kurzreferenz der Bits angezeigt. Durch drücken der Taste „STRG" sowie Klick auf das Register öffnet SiSy eine Hilfedatei mit dem entsprechenden ausführlichen Hilfethema.



Befehlshilfe

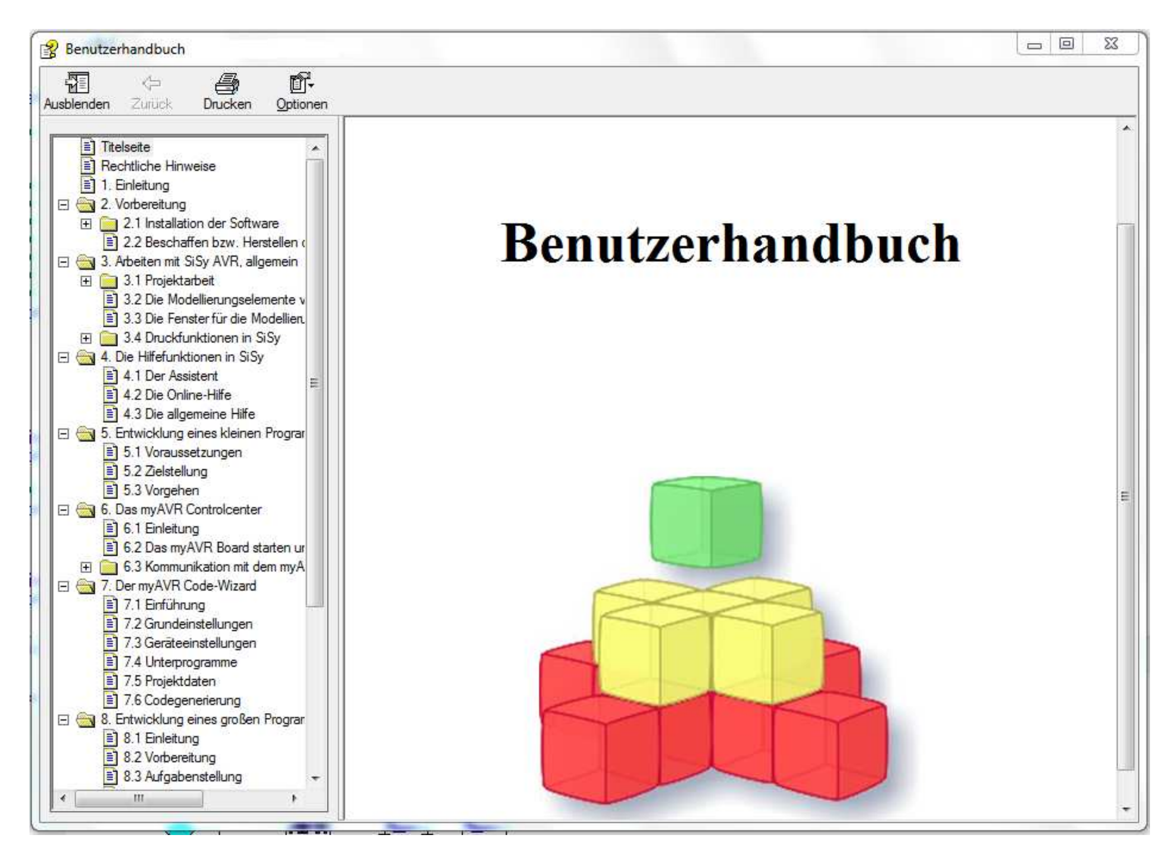
Bei der Eingabe von Befehlen wird in der Regel die Bedeutung bzw. Funktion des Befehls und ein kurzes Syntaxbeispiel eingeblendet. Durch drücken der Taste „STRG“ sowie Klick auf den Befehl öffnet SiSy eine Hilfedatei mit dem entsprechenden ausführlichen Hilfethema.



4.3 Die allgemeine Hilfe

SiSy bietet neben der direkten Hilfe bei der Eingabe von Schlüsselworten auch eine allgemeine Hilfe an.





5 Entwicklung eines kleinen Programms mit SiSy ARM

Schauen wir uns als nächstes kurz in der Entwicklungsumgebung SiSy ARM um. Für die Entwicklung von Mikrocontrollerlösungen bietet sich für den Einstieg die einfache Programmierung (kleines Programm) an.

5.1 Voraussetzungen

Für die Bearbeitung der folgenden Aufgaben benötigen Sie die aufgeführte Software und Hardware.

Software:

- SiSy ARM ab Version 3

Hardware:

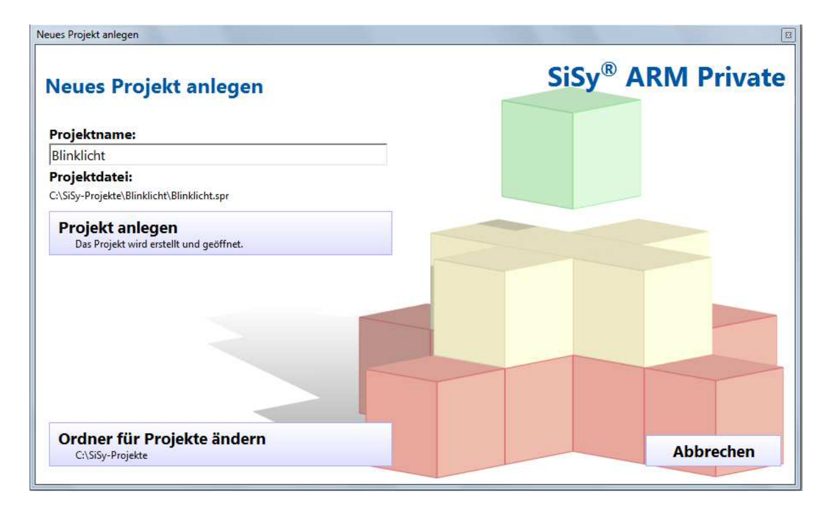
- Ein STM32F4
- Programmierkabel (Mini-USB)
- STM32-Board-F4D
- Patchkabel

5.2 Zielstellung

Der Schnelleinstieg zur Mikrocontroller-Programmierung soll Ihnen helfen, SiSy ARM kennen zu lernen und erste Schritte in der hardwarenahen Programmierung mit SiSy zu gehen.
Zielstellung für das erste Beispielprogramm: Zwei LEDs auf dem STM32F4 sollen in kurzen Zeitabständen aufleuchten und damit ein „Blinklicht“ erzeugen.

5.3 Vorgehen

Ein neues Projekt anlegen

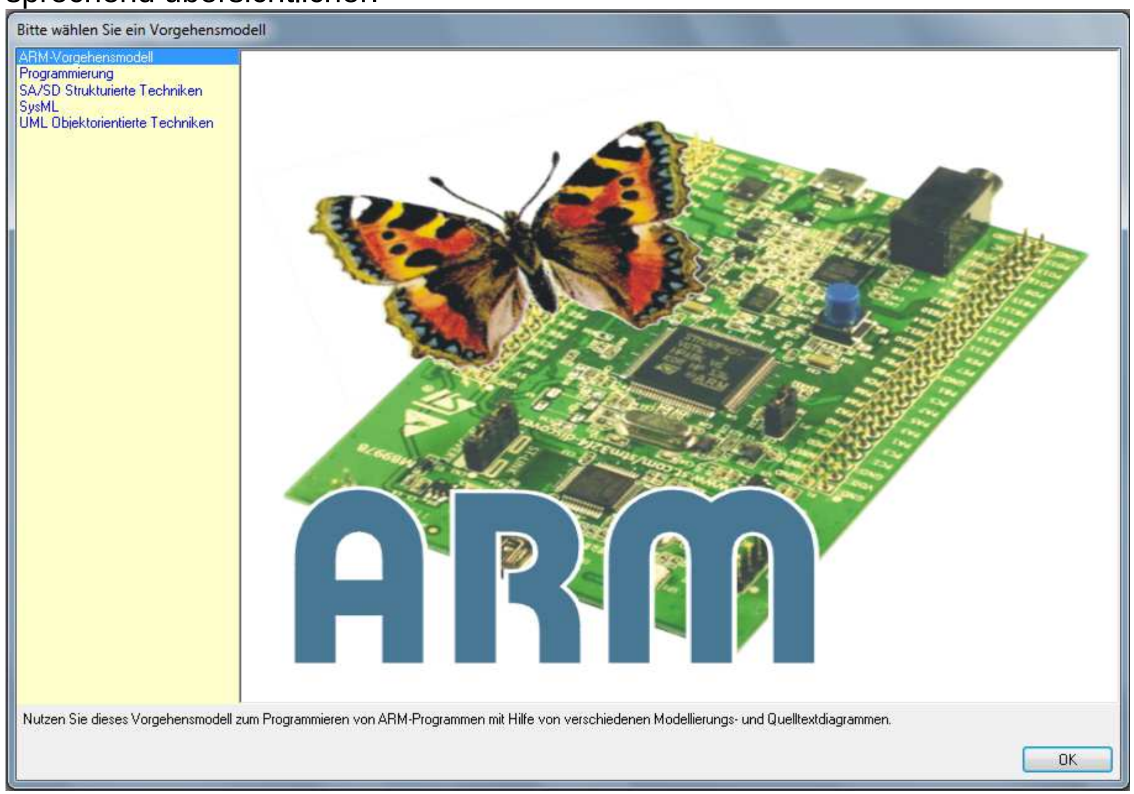


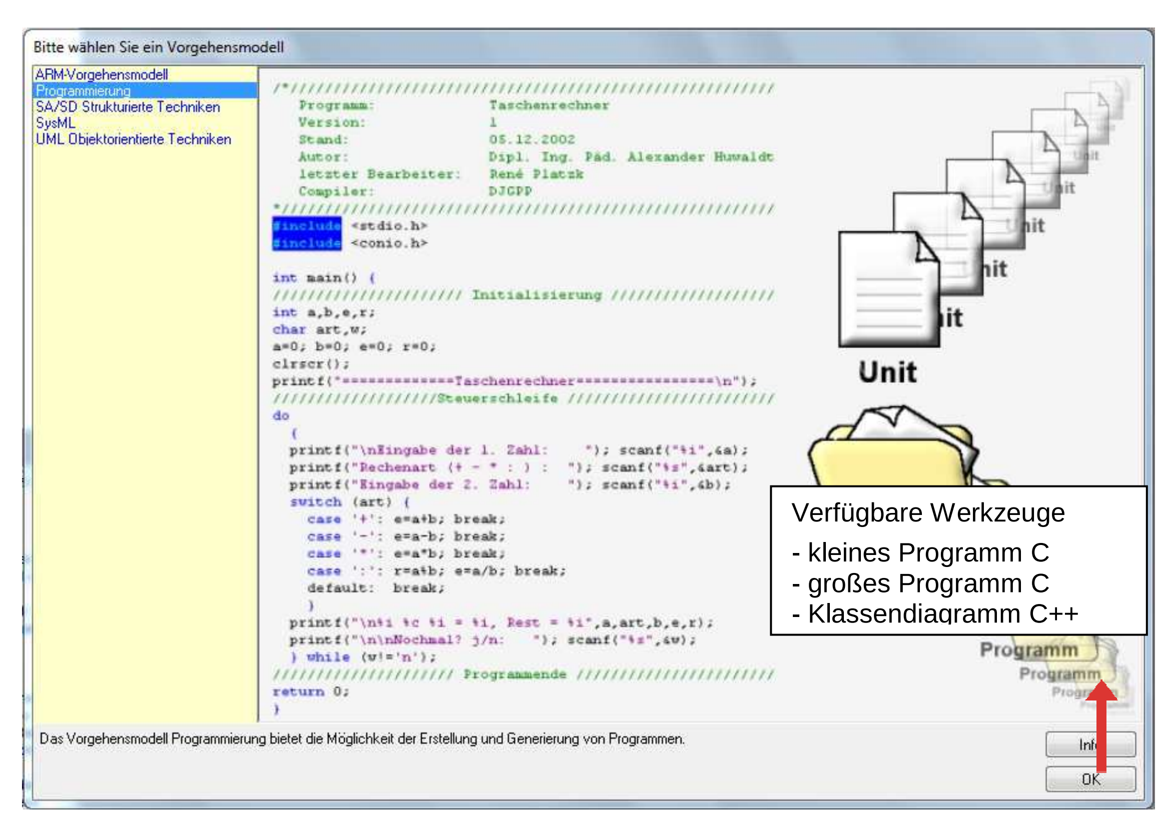
Starten Sie SiSy und wählen Sie „Assistent öffnen“. Danach klicken Sie im SiSy-Assistent auf den Menüpunkt „Neues Projekt anlegen“, vergeben Sie den Projektnamen „Blinklicht“. Bestätigen Sie diesen Dialog mit „Projekt anlegen“.
Wählen Sie das Vorgehensmodell „Programmierung“ und bestätigen Sie mit „OK“.

<u>*Hinweis:*</u>
In SiSy legen Sie stets ein Projekt an. In dieses Projekt integrieren Sie Ihr Programm bzw. mehrere Programme. Unabhängig vom Programmnamen benötigt jedes Projekt einen Namen.
Wenn in ihrer SiSy Ausgabe mehrere Vorgehensmodelle (Hauptebenen) zur Verfügung stehen, wird eine Auswahl der Hauptebene angeboten. Mit der Auswahl des Vorgehensmodells (VGM) entscheiden Sie über das Profil des Projektes. Wenn ein Projekt klein und die Modellierung von Programmablaufplänen nicht erforderlich ist, wählt man das einfachere Vorgehensmodell, in dem gezeigten Beispiel „Programmierung“. Bei

komplexen Projekten, für die neben kleinen Programmen auch Programmablaufpläne, Struktogramme oder Klassendiagramme nötig sind, wird das komplexere Vorgehensmodell „ARM-Vorgehensmodell“ ausgewählt.

Die Auswahl des Vorgehensmodells bestimmt im weiteren Projektverlauf die von SiSy zur Verfügung gestellten Werkzeuge. Bei Auswahl des einfachen Vorgehensmodells „Programmierung“ stehen dann zum Beispiel keine grafischen Werkzeuge wie der Programmablaufplan zur Verfügung. Damit sind die Menüs und Objektbibliotheken entsprechend übersichtlicher.

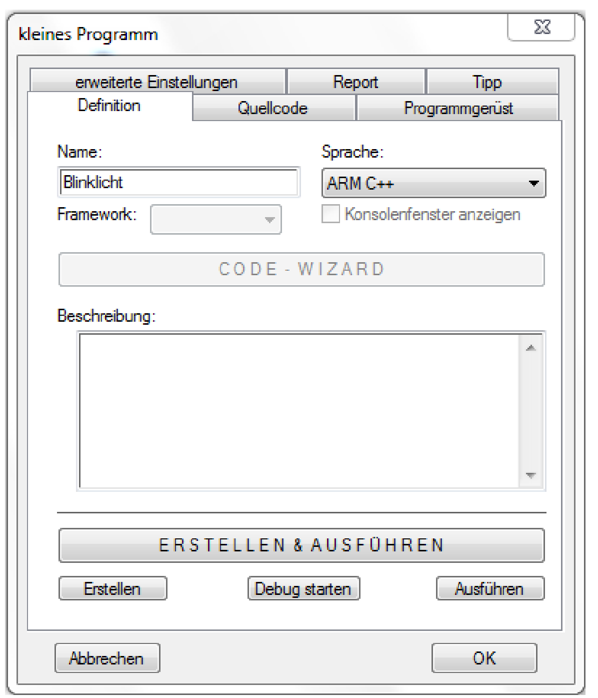




Im weiteren Verlauf werden entsprechend der Auswahl des Vorgehensmodells unterschiedliche Hilfen zum Erstellen eines Programms angeboten. Für Einsteiger empfiehlt sich zuerst die Nutzung des Vorgehensmodells „Programmierung“.
Bei jedem neuen Projekt müssen die Grundeinstellungen zur verwendeten Zielplattform vorgenommen werden (Mikrocontrollertyp, Taktrate, Programmer und I/O-Port).

Kleines C-Programm anlegen

Erstellen Sie ein Programm für den ARM-Mikrocontroller, indem Sie per Drag & Drop aus der Objektbibliothek ein „kleines Programm“ in das Diagrammfenster ziehen. Das Kontextmenü öffnet sich nun automatisch. Für spätere Bearbeitungen, markieren Sie das Objekt und wählen Sie aus dem Kontextmenü (rechte Maustaste) „Definieren“.
Auf der Registerkarte „Definition“ tragen Sie den Programmnamen ein (im Beispiel „Blinklicht“) und wählen die Programmiersprache aus, hier „ARM C++“.



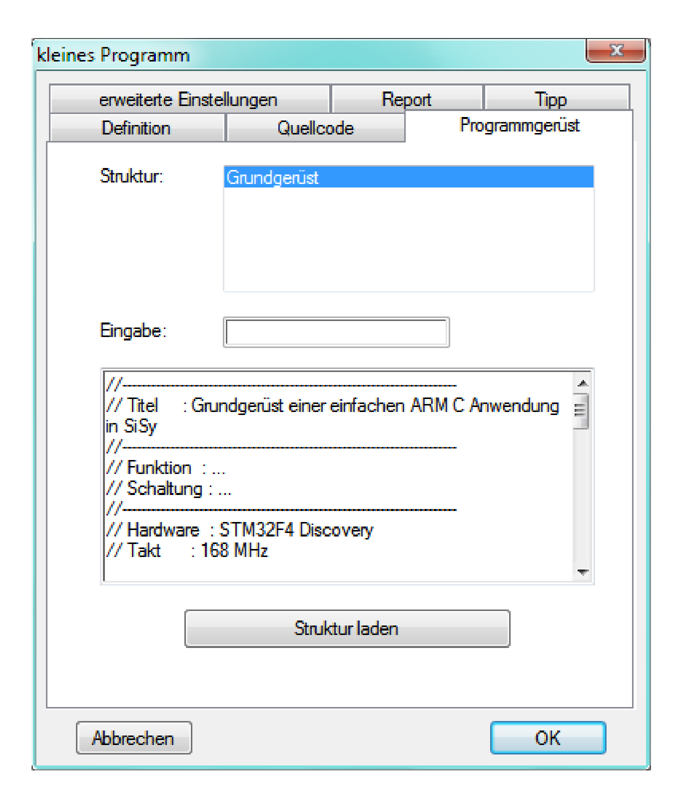
Definition kleines Programm

Programmgerüst laden, Quellcode erstellen

Über die Registerkarte „Programmgerüst“ können Sie das folgende Grundgerüst für ein ARM Programm laden; in der Registerkarte „Quellcode“ können Sie den Quellcode eigenständig eintragen.

Hinweis:
Den Quellcode können Sie auch im Beschreibungsfenster/Editorfenster der SiSy-Benutzeroberfläche eintragen bzw. korrigieren.

Laden Sie die Vorlage für das Programmgerüst oder erstellen Sie den folgenden Quellcode.
Vergleichen Sie dazu auch den Abschnitt zum myAVR Code-Wizard (Abschnitt 0).
Das Programmgerüst darf erst geladen werden, wenn der Zielcontroller ausgewählt wurde; die Vorlagen sind controllerspezifisch.



Grundgerüst laden

```
#include <stddef.h>
#include <stdlib.h>
#include "stm32f4xx.h"

void initApplication()
{
    SysTick_Config(SystemCoreClock/100);
    // weitere Initialisierungen durchführen
}

int main(void)
{
    SystemInit();
    initApplication();
    do{
        // Eingabe
        // Ausgabe
        // Verarbeitung
    } while (true);
    return 0;
}

extern "C" void SysTickFunction(void)
{
    // Application SysTick

}
```

Quellcode in C erstellen

Das Blinklicht soll über die LEDs angezeigt und von dem Prozessorport GPIOD gesteuert werden. Die Realisierung erfolgt über GPIO Pin 12 und 13. Dieses wird mit dem Befehl Bit-Rotation nach rechts verschoben und an den Port GPIOD des Prozessors ausgegeben. Auf Grund der Prozessorgeschwindigkeit muss die Ausgabe des Blinklichtes für unser Auge verzögert werden. Geben Sie folgenden Quellcode ein bzw. ergänzen Sie die Programmvorlage!

```
#include <stddef.h>
#include <stdlib.h>
#include "stm32f4xx.h"

void initApplication()
{
    SysTick_Config(SystemCoreClock/100);
    // weitere Initialisierungen durchführen

    /* GPIOG Periph clock enable */
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);

    /* Configure PG6 and PG8 in output pushpull mode */
    GPIO_InitTypeDef  GPIO_InitStructure;
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_12 | GPIO_Pin_13;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIOD, &GPIO_InitStructure);
}
```



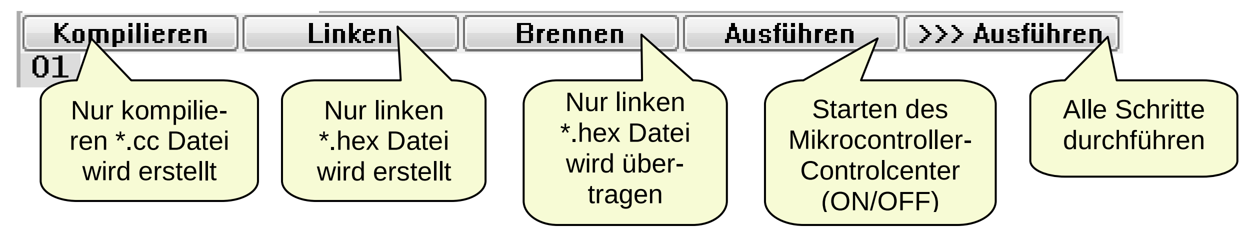
```
void delay(vu32 nCount)
{
      while(nCount--);
}

int main(void)
{
    SystemInit();
    initApplication();
    GPIO_SetBits(GPIOD,GPIO_Pin_13);
    do{
        // Eingabe
        // Ausgabe
        // Verarbeitung
        GPIO_ToggleBits(GPIOD,GPIO_Pin_13|GPIO_Pin_12);
        delay(10000000);
        } while (true);
    return 0;
}

extern "C" void SysTickFunction(void)
{
    // Application SysTick
}
```

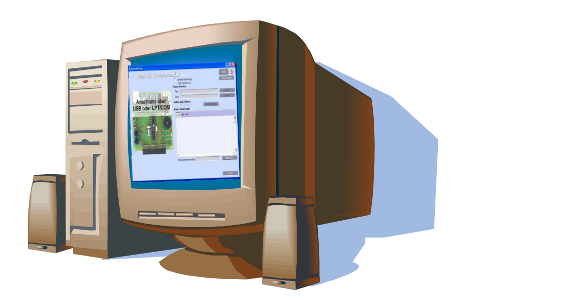
Kompilieren und Linken

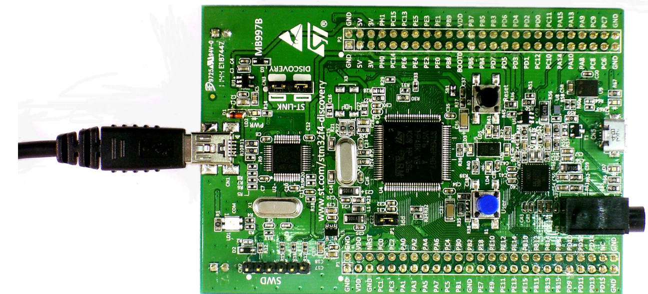
Der eingegebene Quellcode muss nun in Maschinencode für den ARM-Prozessor übersetzt werden. Wählen Sie dazu die Schaltflächen „Kompilieren“ und „Linken“. Bei fehlerfreier Übersetzung liegt das Programm als „Blinklicht.hex“ vor und kann auf den FLASH-Programmspeicher des Prozessors gebrannt werden.

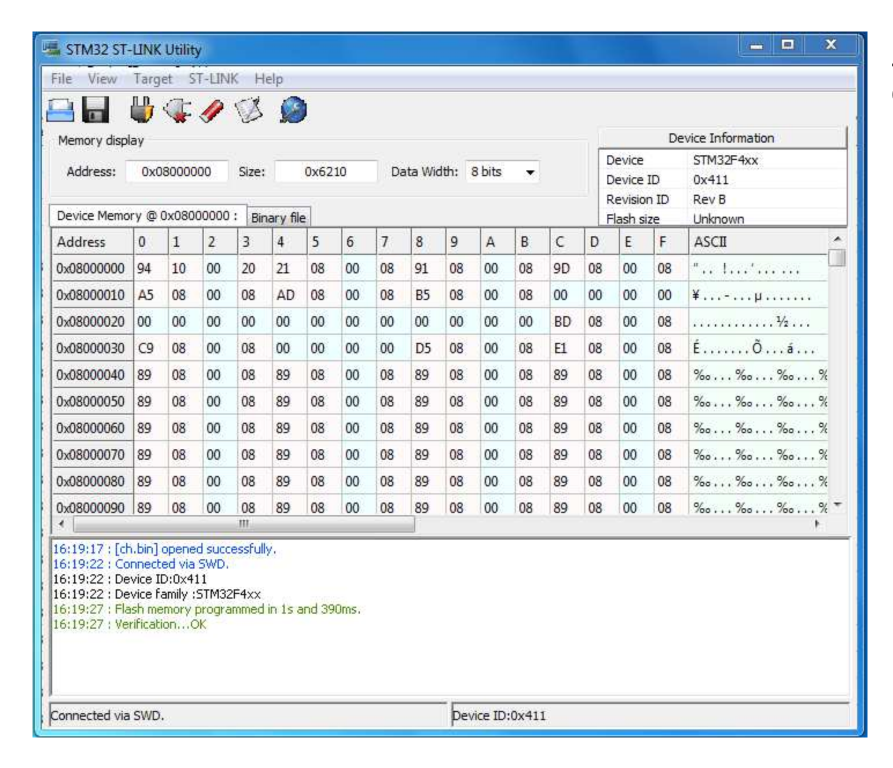


Hardware anschließen und brennen

Das STM32F4 verfügt über eine ISP (In System Programming) Schnittstelle. Der Prozessor muss also nicht für die Programmierung aus dem System entfernt werden, um ihn in einem gesonderten Programmiergerät zu brennen, sondern kann im STM32F4 direkt programmiert werden. Dazu schließen Sie das Programmierkabel an den USB-Port Ihres Rechners an.

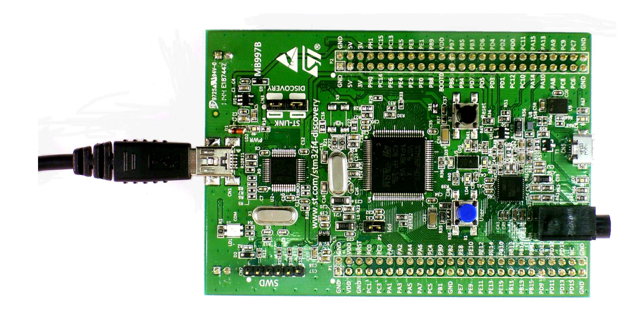






Zum Brennen wählen Sie die Schaltfläche „Brennen“. Bei erfolgreichem Brennvorgang erhalten Sie im Ausgabefenster vom ST-Link folgende Meldung:

Mikrocontrollerlösung testen



Um das Programm zu testen ist es nötig, dass das STM32F4-Discovery angeschlossen ist.

- Es ist an den LEDs ein Blinklicht zu sehen.
- Gratulation!
  Das ist Ihre erste Mikrocontrollerlösung mit dem Board.

Beim Kompilieren, Linken und Brennen des Schnellstart-Beispiels öffnet sich ein Ausgabefenster und zeigt Protokollausgaben der Aktionen an. Beim Brennen öffnet sich zusätzlich das ST-Link. Wenn die Hardware ordnungsgemäß angeschlossen, von der Software erkannt und das Programm erfolgreich auf den Programmspeicher des Mikrocontrollers übertragen wurde, schließen Sie ST-Link. Die letzte Ausschrift hat folgenden Inhalt:

```
16:19:17 : [ch.bin] opened successfully.
16:19:22 : Connected via SWD.
16:19:22 : Device ID:0x411
16:19:22 : Device family :STM32F4xx
16:19:27 : Flash memory programmed in 1s and 390ms.
16:19:27 : Verification...OK
```

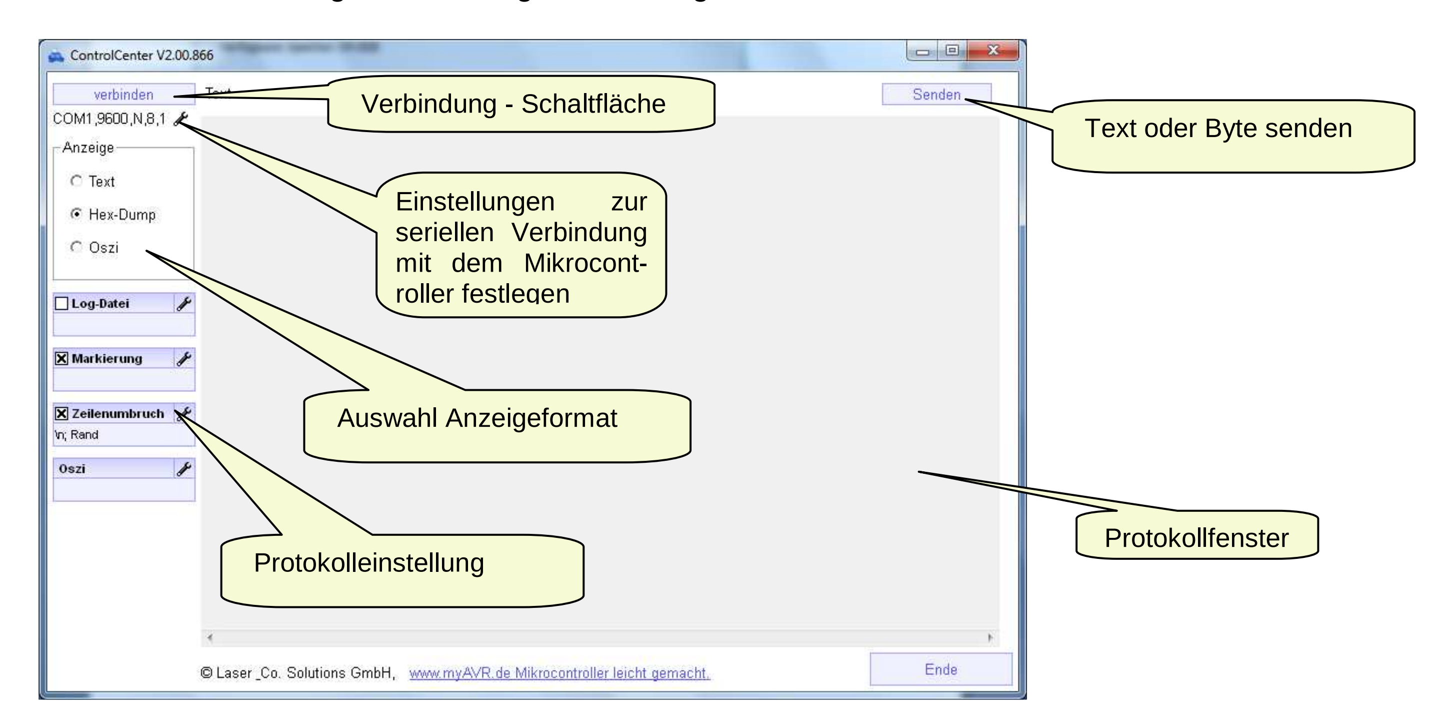
Ausgabefenster mit "Brenn"-Protokoll

Inbetriebnahme, Test und Datenkommunikation mit der Mikrocontrollerlösung erfolgen über das myAVR Controlcenter. Dabei wird über die Schaltfläche „Start“ das Board mit der nötigen Betriebsspannung versorgt und der Controller gestartet. Der Datenaustausch mit dem myAVR Board ist möglich, wenn das Null-Modemkabel an Rechner und Board angeschlossen ist, sowie die Mikrocontrollerlösung dafür vorgesehen ist. Es können Texte und Bytes (vorzeichenlose ganzzahlige Werte bis 255) an das Board gesendet und Text empfangen werden. Die empfangenen Daten werden im Protokollfenster angezeigt. Vergleichen Sie dazu den folgenden Abschnitt zum myAVR-Controlcenter.

6 Das Mikrocontroller Controlcenter

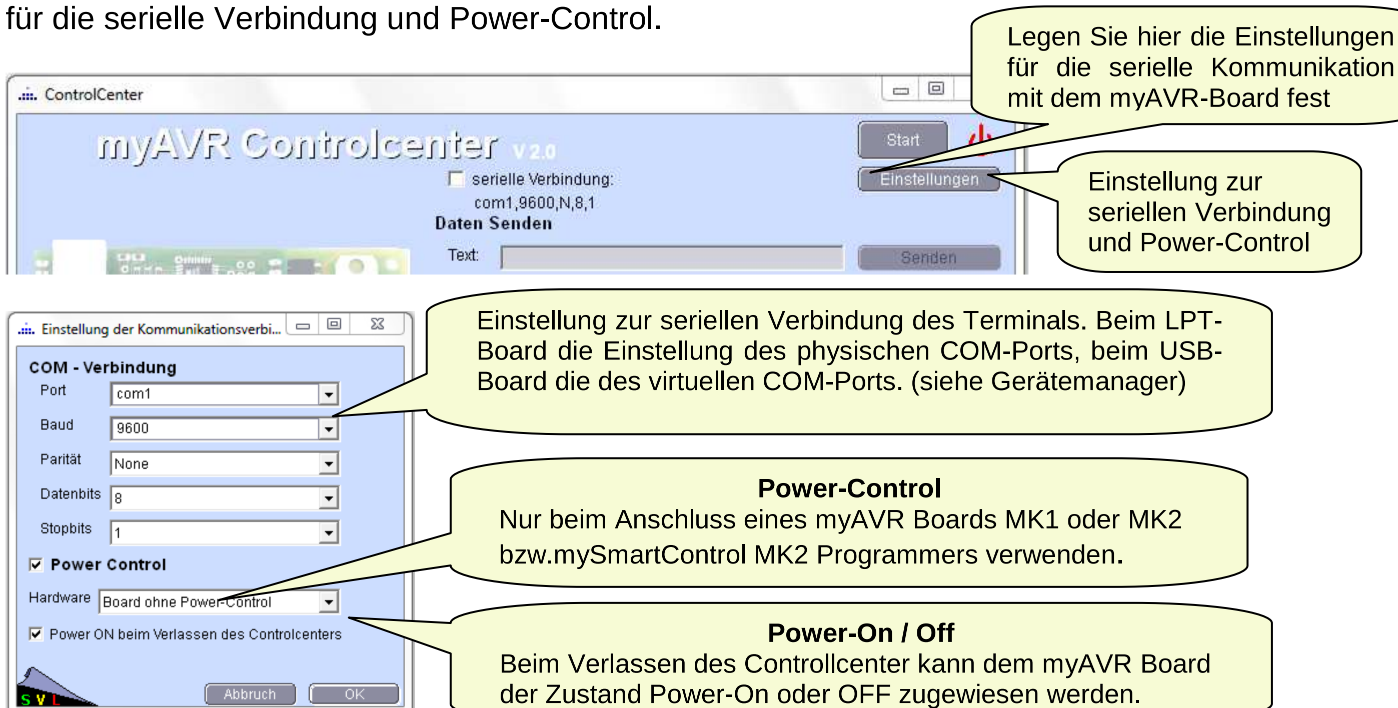
6.1 Einleitung

Das Mikrocontroller Controlcenter ist ein universelles Terminalprogramm zur Kommunikation mit dem Mikrocontroller Board und anderen Mikrocontrollerapplikationen, die über eine serielle Schnittstelle (UART) oder USB Anbindung mit virtuellem COM-Port zum PC verfügen. Es kann für Test- und Debug-Meldungen sowie Visualisierung und Protokollierung von Messdaten genutzt werden. Dazu bietet das Mikrocontroller Controlcenter umfangreiche Konfigurationsmöglichkeiten.

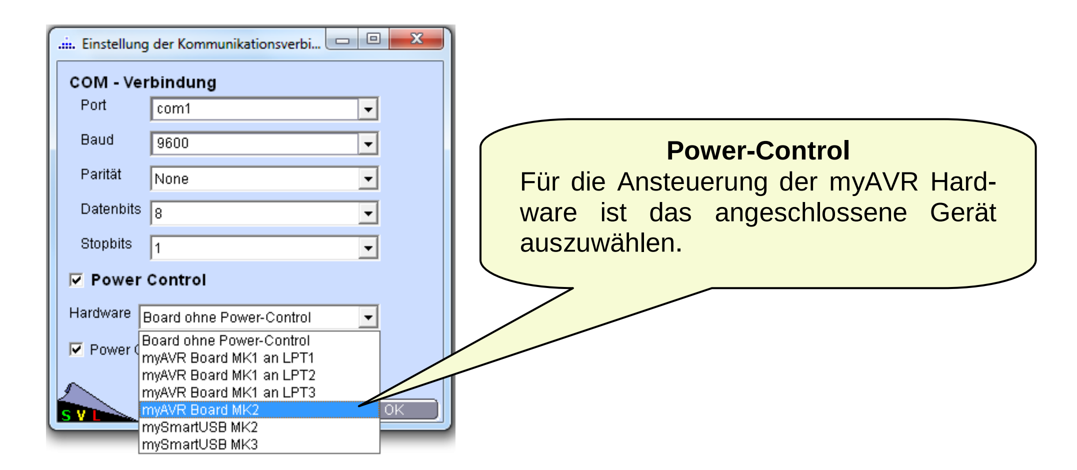


6.2 Das STM32F4 Board starten und stoppen (Power Control)

Diese Einstellung ist speziell für myAVR Programmer und muss für ARM deaktiviert werden. Wenn Sie ein STM32F4 Board besitzen, können Sie die Spannungsversorgung des Boards aus dem Controlcenter heraus steuern. Dazu wählen Sie die Einstellungen für die serielle Verbindung und Power-Control.



Mit dem Aktivieren der Power-Control-Funktion wird beim Betätigen der Schaltfläche „Start/Stopp“ die rechnerseitige Spannungsversorgung des Mikrocontroller Boards und die RESET-Leitung angesteuert.



Bei einer externen Spannungsversorgung, zum Beispiel durch einen 9V Block oder das Netzteil, hat die Power-Control-Funktion keine Wirkung auf die Boardspannung. In diesem Fall bezieht sich die Wirkung ausschließlich auf die RESET-Leitung, um den Controller zu starten bzw. zu stoppen. Beachten Sie, dass beim myAVR Board MK1 die Spannungsversorgung über den LPT-Port für eine serielle Kommunikation und den Betrieb des LC-Displays nicht ausreicht, dafür wird die Verwendung eines myAVR Netzteils empfohlen.

6.3 Kommunikation mit dem STM32 F4 Discovery PLUS

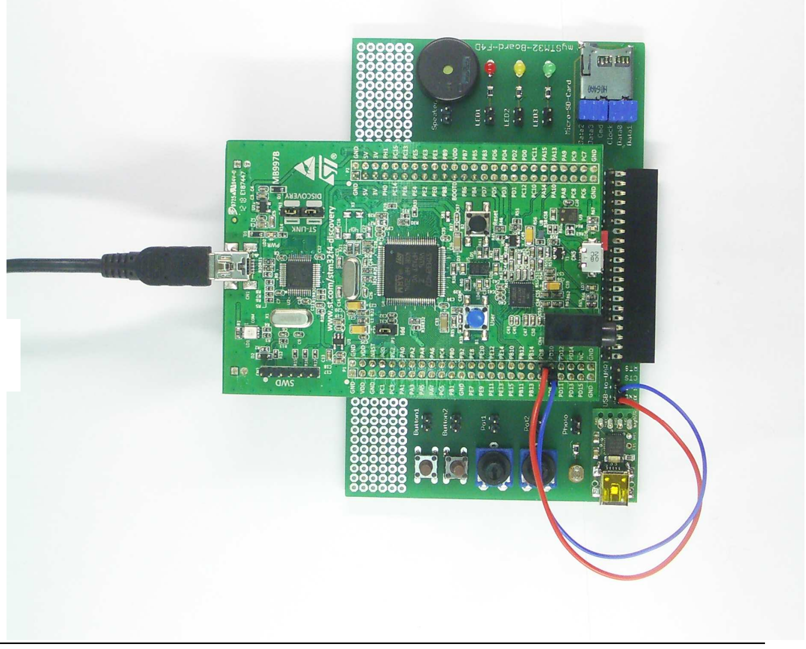
6.3.1 Grundlagen (USB-Variante)

Das STM32F4-Discovery verfügt über einen Mini-USB Anschluss, dieser wird für die Stromzufuhr genutzt. Am mySTM32-Board-F4D befindet sich ein weiterer Mini-USB Anschluss zur Kommunikation mit den Boards, die durch Patchkabel verbunden sind.

Die Patchkabel sind wie folgt anzuschließen:

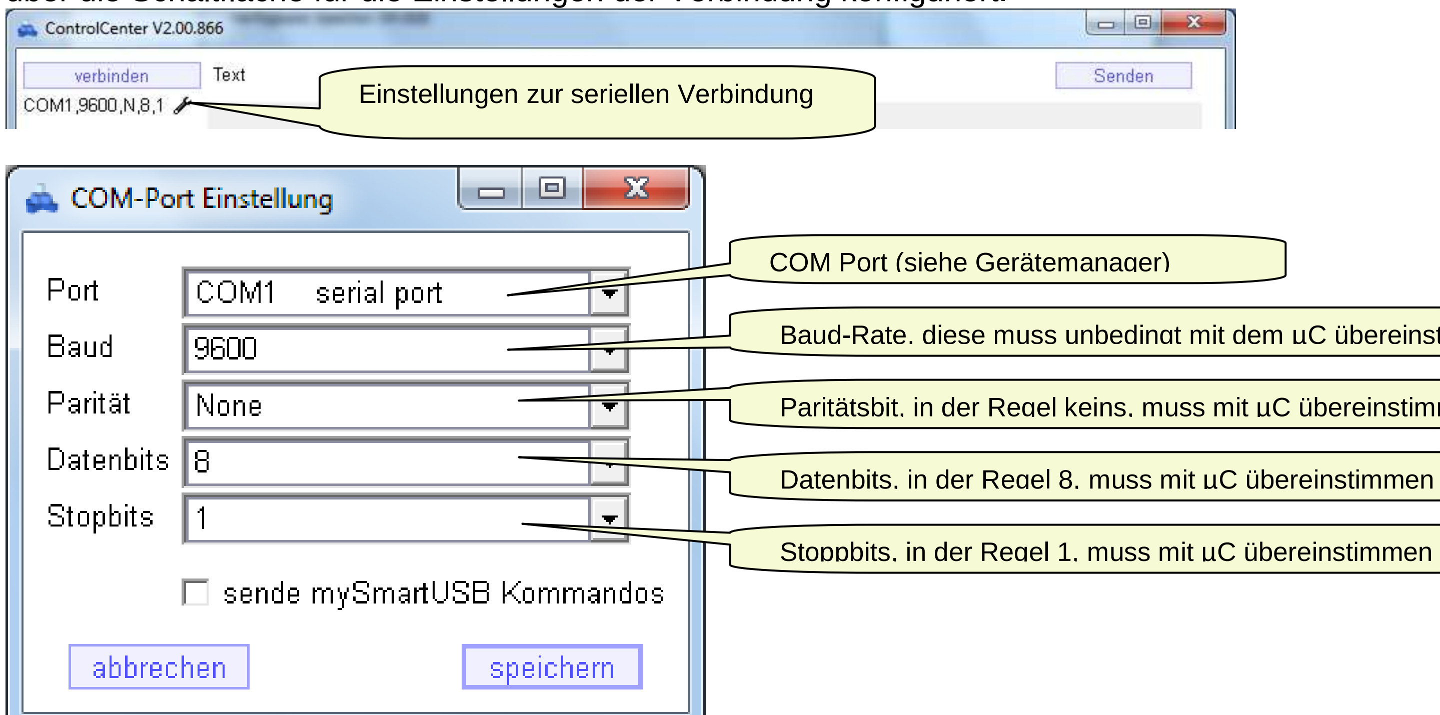
1. am mySTM32-Board-F4D *RxD* und am STM32F4-Discovery *PD8*
2. am mySTM32-Board-F4D *TxD* und am STM32F4-Discovery *PD9*



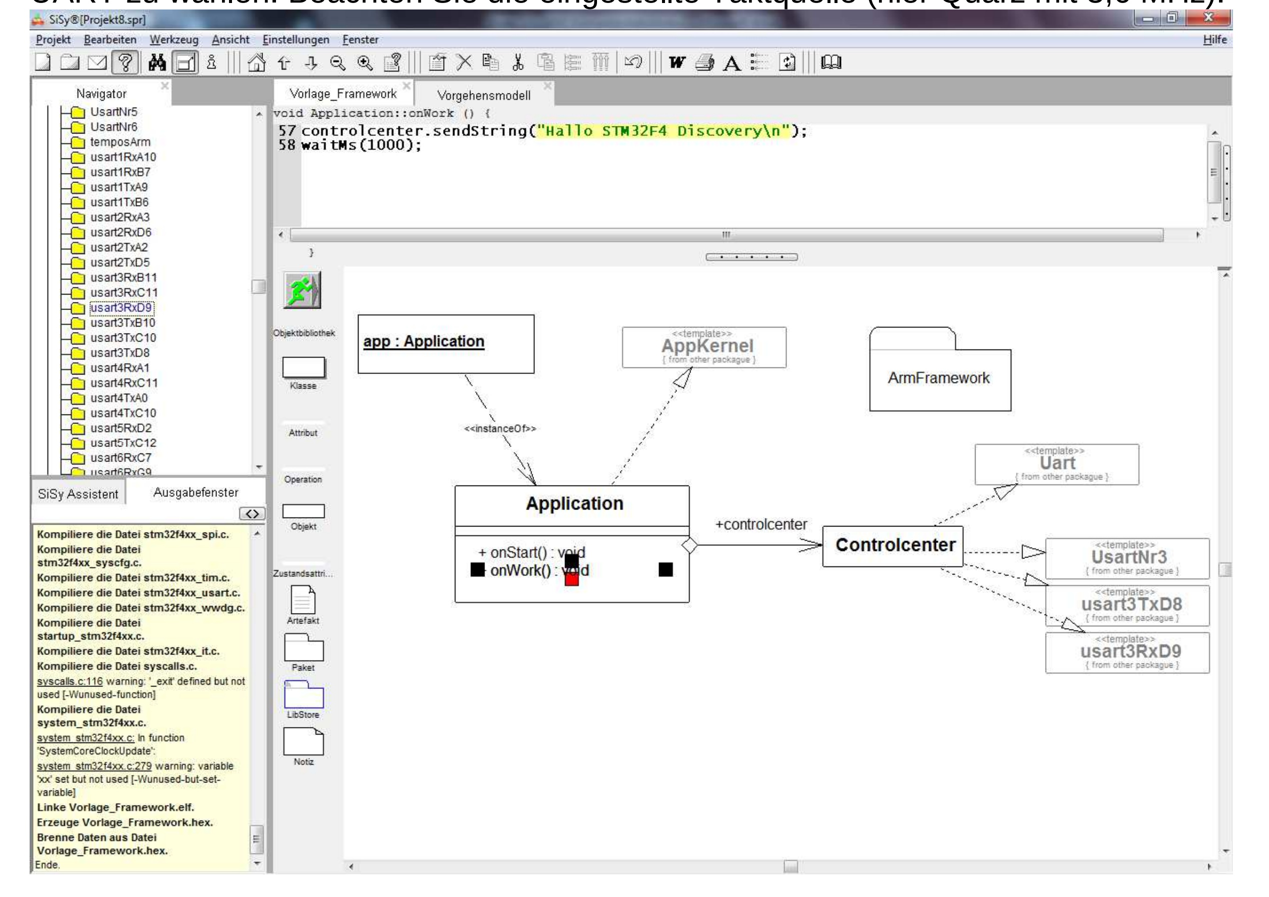


6.3.2 Einstellungen für die seriellen Verbindung

Für eine erfolgreiche Kommunikation mit dem STM32F4 ist es wichtig, dass Sender und Empfänger von seriellen Daten die gleichen Parameter für die Datenübertragung konfiguriert haben. Auf der PC Seite werden die Kommunikationsparameter im Controlcenter über die Schaltfläche für die Einstellungen der Verbindung konfiguriert.

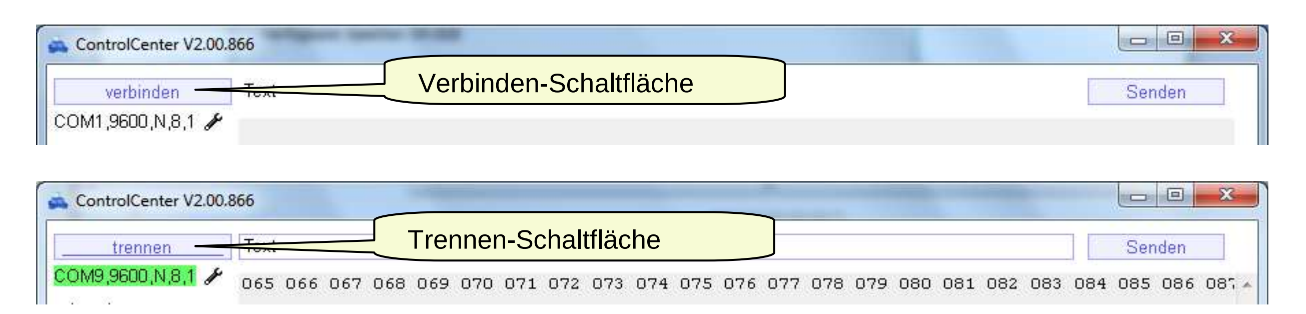


Im Mikrocontrollerprogramm sind die gleichen Parameter bei der Initialisierung der UART zu wählen. Beachten Sie die eingestellte Taktquelle (hier Quarz mit 3,6 MHz).



6.3.3 Daten empfangen vom STM32F4 Board

Das Mikrocontroller Controlcenter empfängt Daten über den gewählten COM-Port und stellt diese im Protokollfenster dar. Damit können Statusmeldungen, Fehlermeldungen oder auch Messwerte erfasst werden. Die Voraussetzung ist, dass die serielle Verbindung hergestellt (Mini-USB oder USB Kabel), korrekt konfiguriert und aktiviert wurde. Die Kommunikation beginnt mit dem Betätigen der Schaltfläche „Verbinden" und endet mit dem Betätigen der Schaltfläche „Trennen". Der Zustand (in Betrieb) wird bei der seriellen Verbindung angezeigt, indem diese (grün) hinterlegt wird, links unter der Verbinden-Schaltfläche.

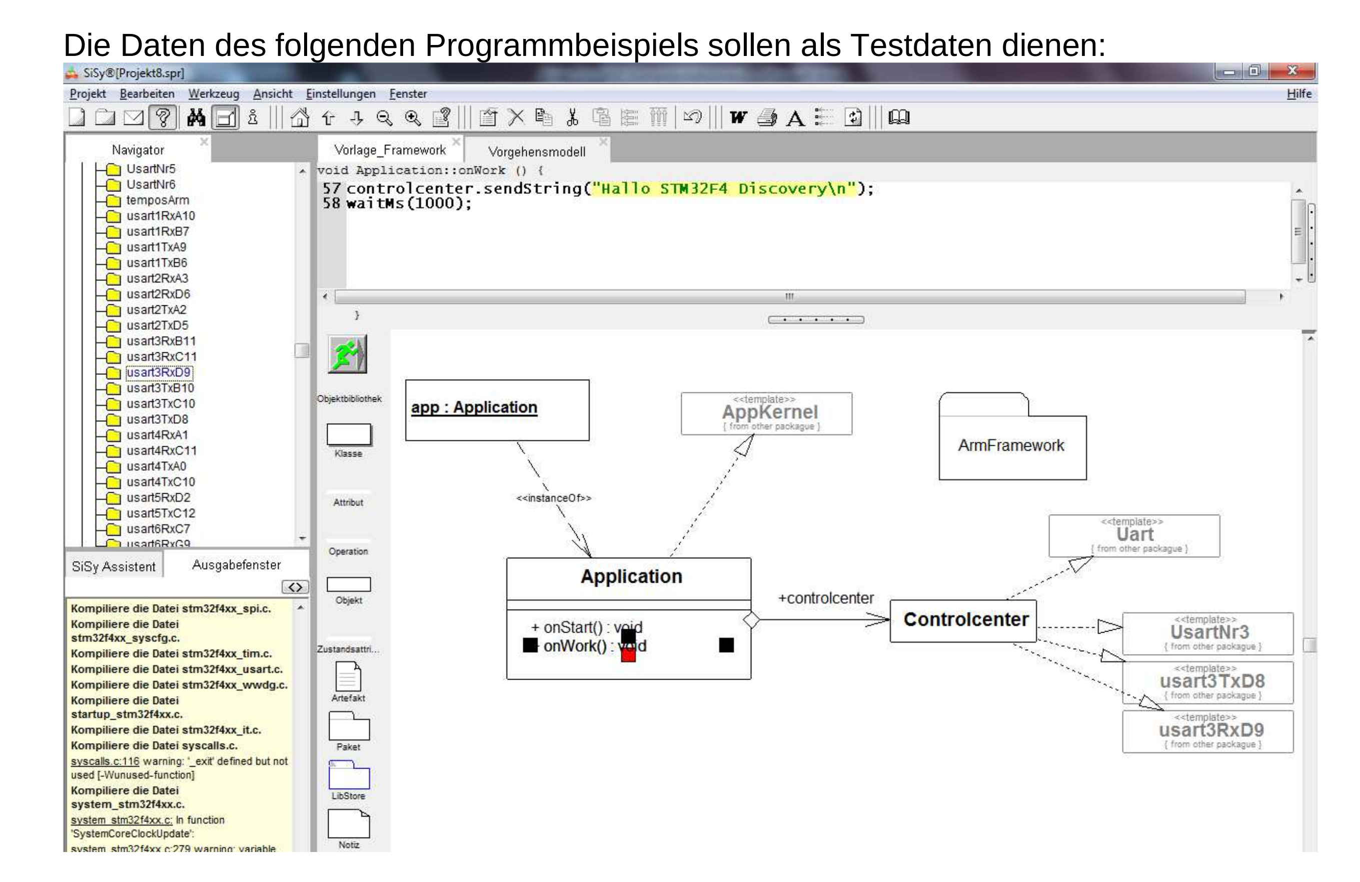


6.3.4 Darstellung der empfangen Daten

Die empfangenen Daten werden fortlaufend im Protokollfenster dargestellt. Der Darstellungsmodus kann während der Kommunikation umgeschaltet werden. Das Controlcenter bietet folgende Darstellungsmodi:

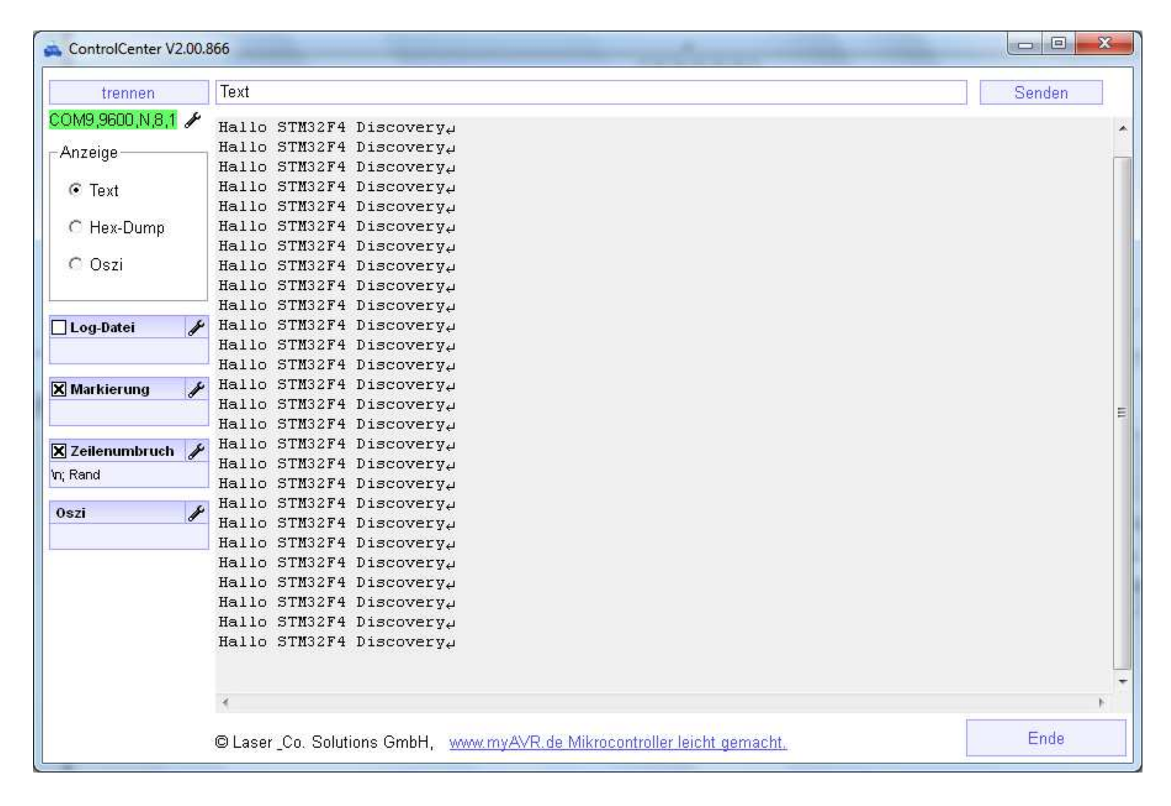
- Text
- Zahl
  - o Dezimal
  - o Hexadezimal
- Grafik (Oszi)

Die Daten des folgenden Programmbeispiels sollen als Testdaten dienen:

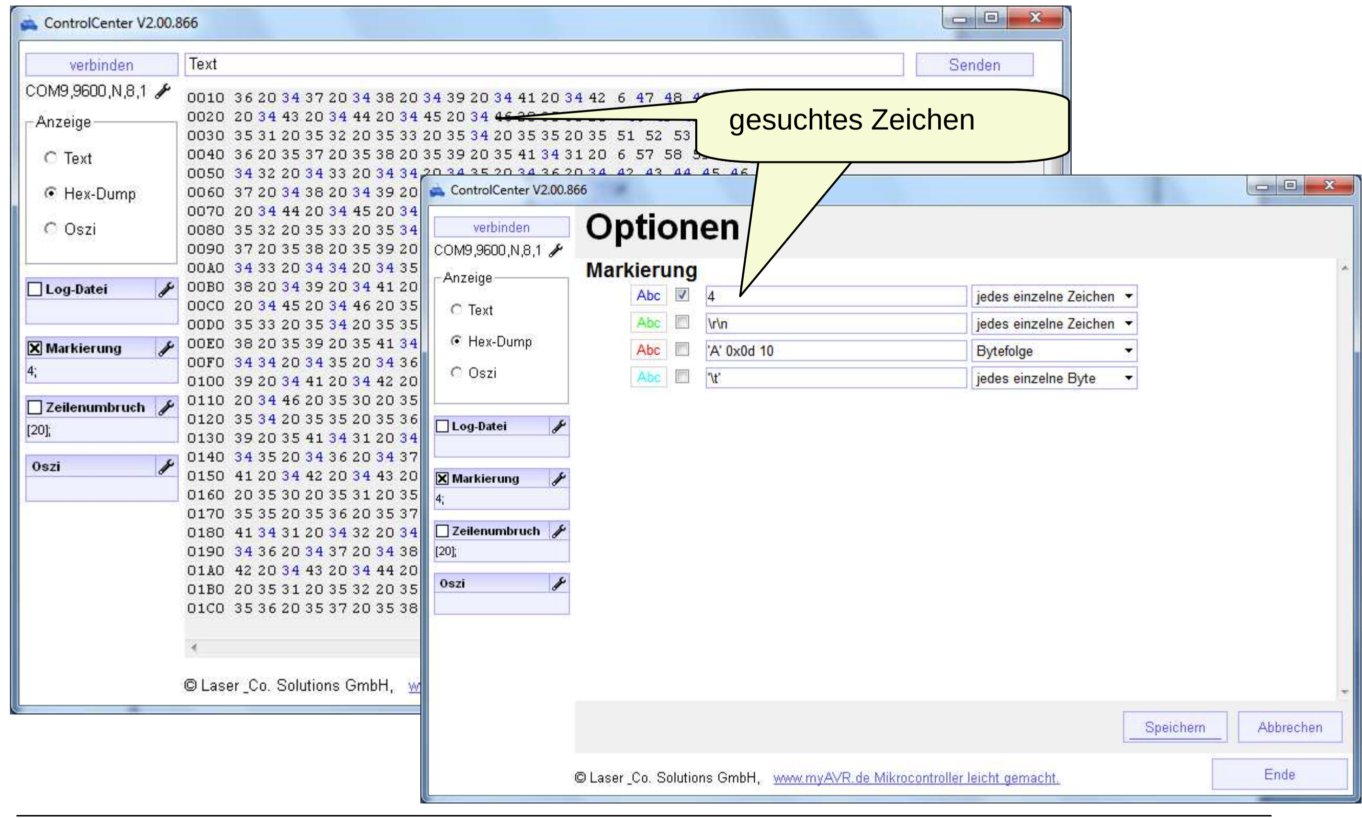


Der Textmodus

Der Textmodus dient zur Visualisierung alphanumerischer Werte im ASCII Format (Zeichenketten). Zahlen, die mit Zeichenketten gesendet werden, müssen vom Mikrocontrollerprogramm zuvor ins ASCII-Format gewandelt werden (siehe `itoa` und `sprintf`).



Über die Schaltflächen auf der linken Seite, lässt sich das Protokollfenster weiter konfigurieren. Die Zeilenbreite und das Zeichen für einen Zeilenumbruch lassen sich auswählen. Es können wichtige/gesuchte Textstellen im Protokollfenster hervorgehoben werden.

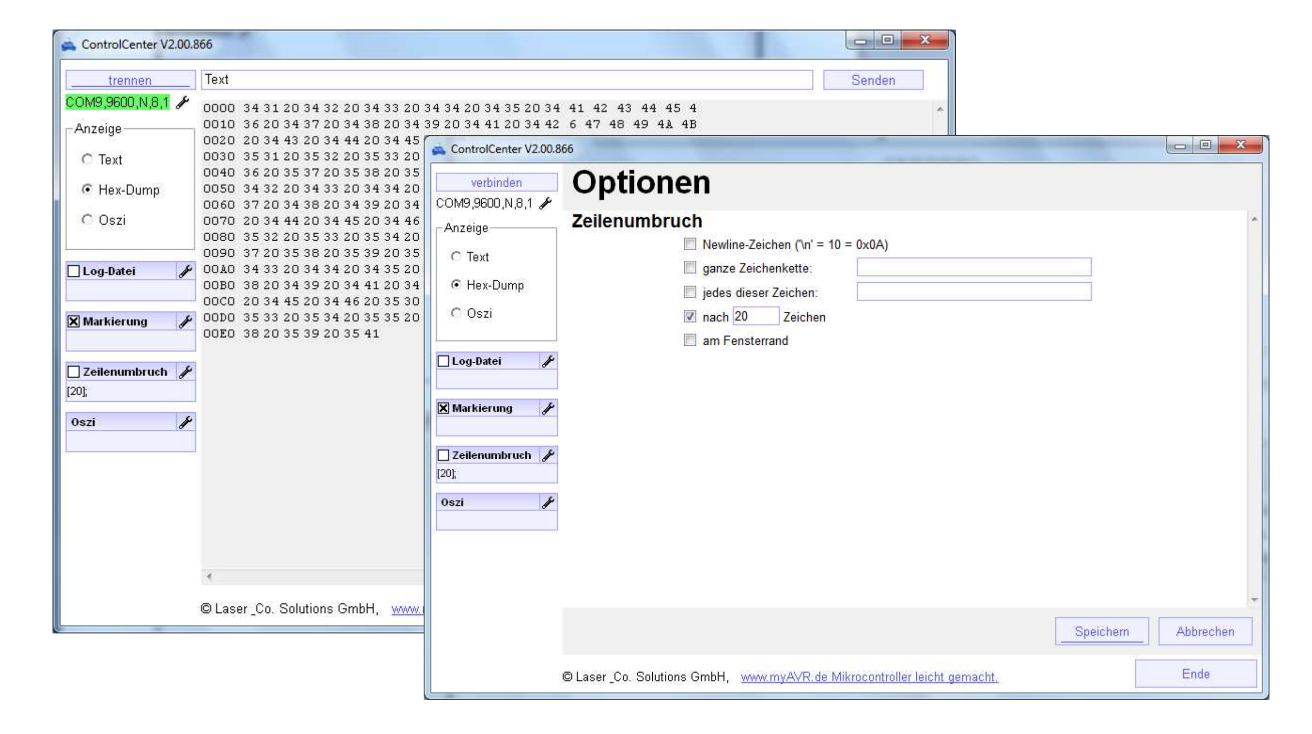


Der Zahlenmodus

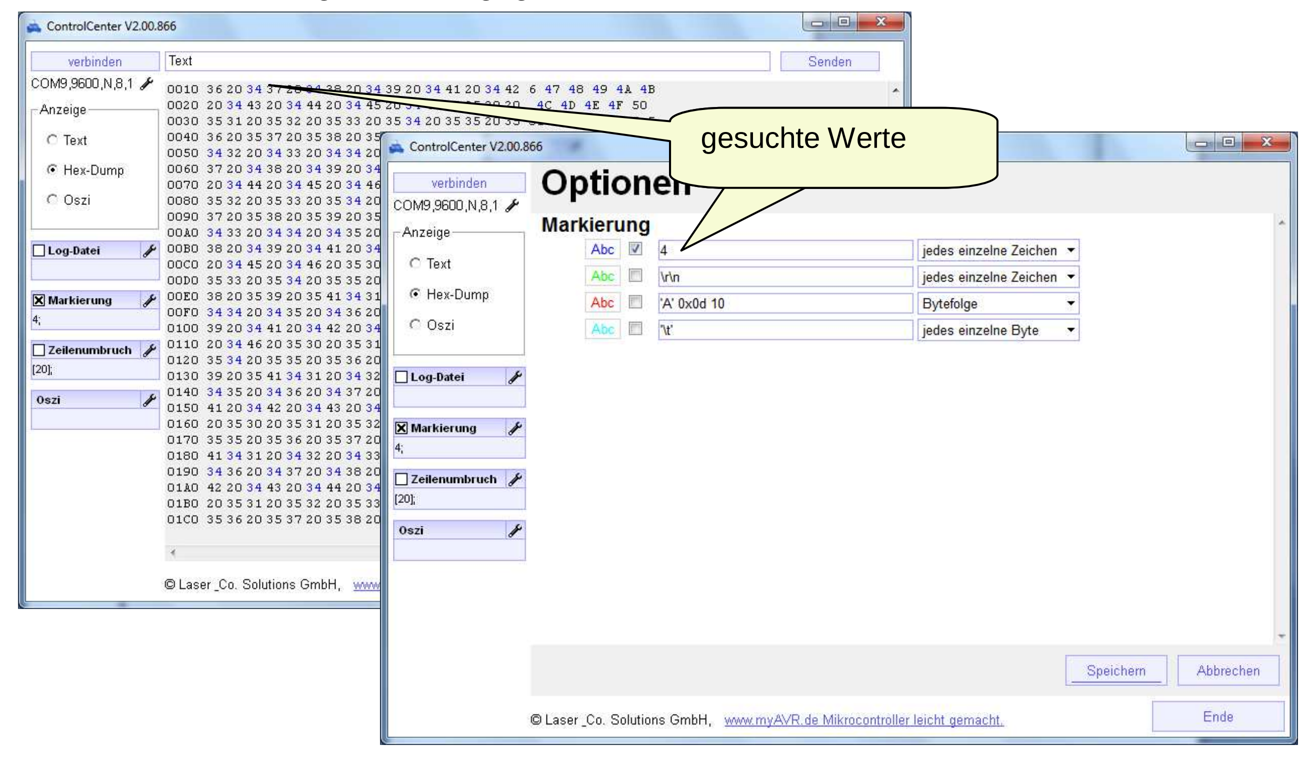
Der Zahlenmodus visualisiert die empfangenen Datenbytes (8 Bit, Werte von 0 bis 255) als Dezimal- oder Hexadezimalzahlen. Dezimalzahlen werden dreistellig mit führender Null dargestellt.



Die Anzeige kann über die Schaltflächen auf der linken Seite weiter konfiguriert werden. Dabei werden die Zeilengröße oder das Zeilenumbruchzeichen sowie das Format der Zahlendarstellung (Hexadezimal/Dezimal) ausgewählt.

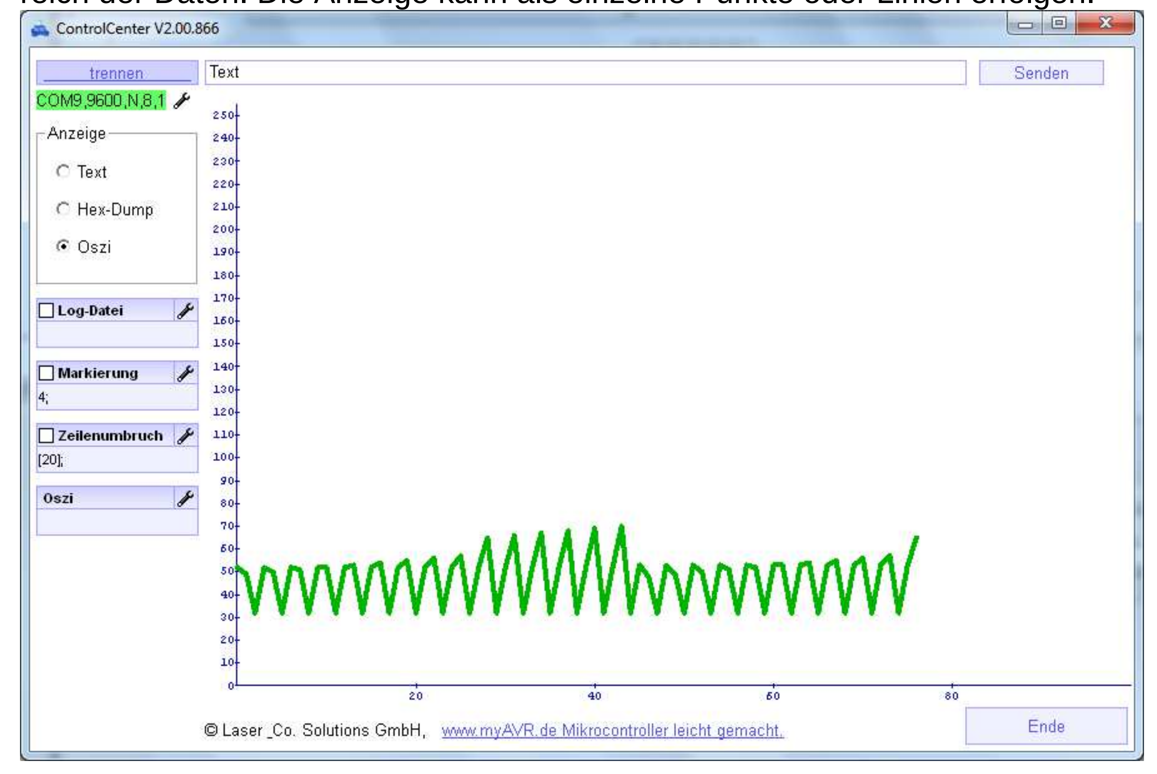


Es können wichtige/gesuchte Zahlenwerte im Protokollfenster hervorgehoben werden. Hexadezimale Zahlen sind durch den Präfix 0x zu kennzeichnen. Mehrere Werte können mit Leerzeichen getrennt angegeben werden.

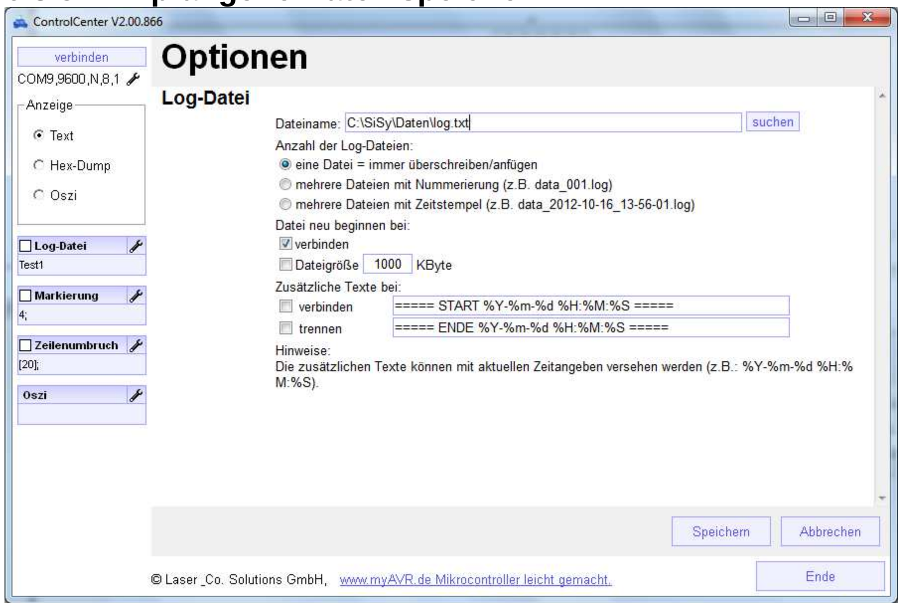


Der Grafikmodus

Messwerte können auch grafisch visualisiert werden. Dabei werden die Werte fortlaufend und byteweise (Wertebereich 0 bis 255) als Punkte in einem Koordinatensystem visualisiert. Die X-Achse repräsentiert den zeitlichen Verlauf, die Y-Achse den Wertebereich der Daten. Die Anzeige kann als einzelne Punkte oder Linien erfolgen.



6.3.5 Empfangene Daten speichern



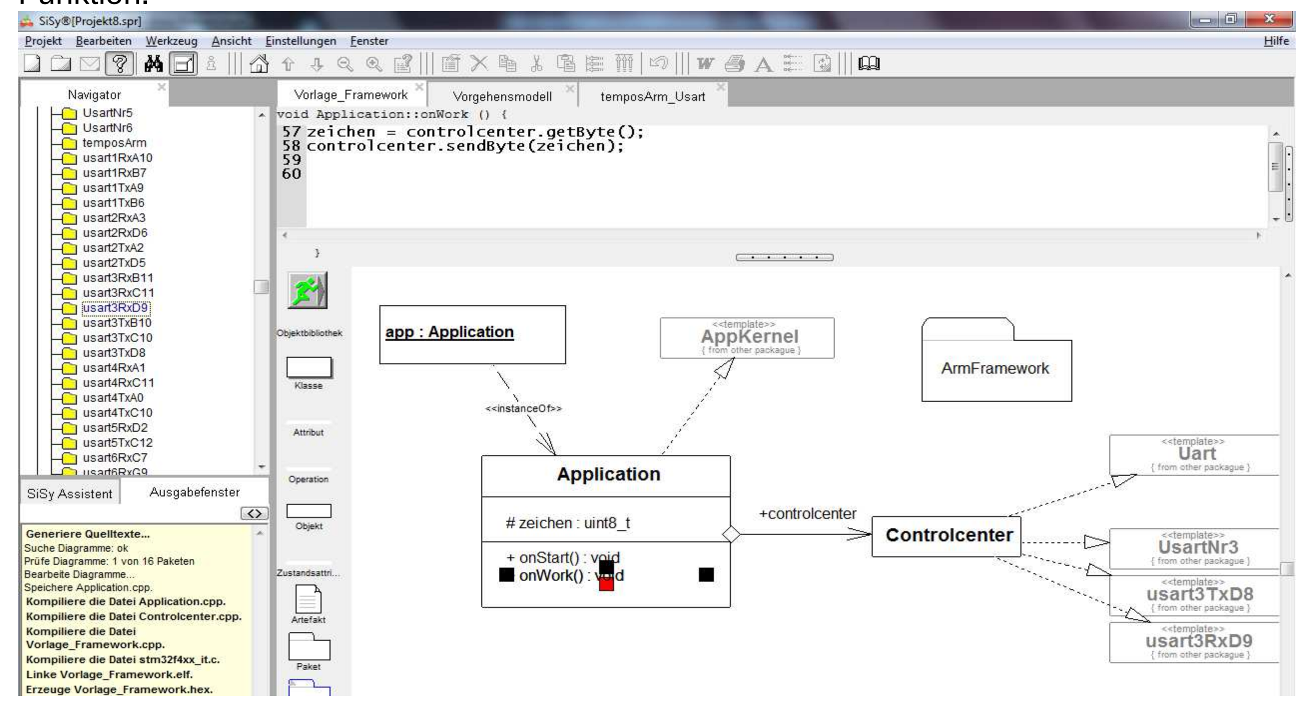
Bei der Erfassung von Messdaten die vom Mikrocontroller an den PC gesendet werden ist es oft wichtig, diese in eine Datei zu speichern. Damit wird eine Weiterverarbeitung der Daten in entsprechenden Programmen (z.B.: EXCEL) möglich. Das Microcontroller Controlcenter ermöglicht es, Protokolldaten aufzuzeichnen (Log-Datei, Rekorder-funktion). Um diese Funktion zu aktivieren, wählen Sie die Option „Log-Datei".

Dort muss der Dateiname und Pfad der Log-Datei angegeben werden. Zusätzlich ist es möglich, das Format und die Blockgröße (Zeilenumbruch) festzulegen. Die Weiterverarbeitung der Daten erfolgt entsprechend der Möglichkeiten der Zielanwendung (z.B.: Öffnen, Importieren oder Zwischenablage).

öffnen mit Exel

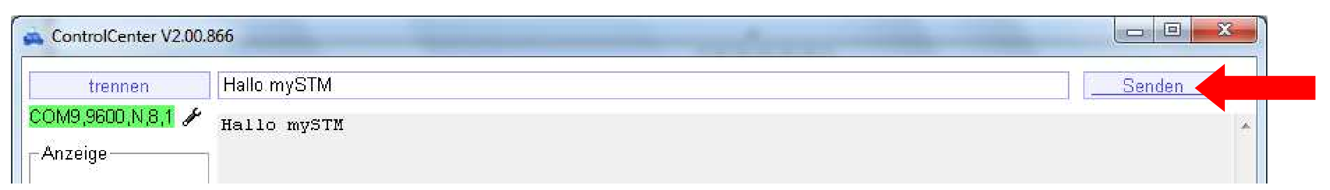
6.3.6 Daten an das Board senden

Über das Controlcenter können Daten im Text- oder Zahlenformat an den Mikrocontroller gesendet werden. Das folgende Programm dient als Veranschaulichung dieser Funktion.



Text senden

Es können einzelne Zeichen, aber auch Zeichenketten gesendet werden. Mit der zugehörigen Schaltfläche „Senden" wird immer der gesamte Inhalt der Eingabezeile gesendet.



Zahlen senden

Zahlenwerte von 0 bis 255 (1 Byte) können einzeln oder als Zahlenfolge gesendet werden. Zwischen den Werten ist ein Leerzeichen als Trennzeichen einzufügen. Zahlen, die größer sind als 255, werden auf den niederwertigen Teil gekürzt. Es ist möglich, Zahlen im Hexadezimalformat zu schreiben (Präfix 0x). Mit der zugehörigen Schaltfläche „Senden" wird immer der gesamte Inhalt der Eingabezeile gesendet.



7 Entwicklung eines großen Programms mit SiSy ARM

7.1 Einleitung

Für die Entwicklung eines größeren Programms ist es unzweckmäßig, alle Befehle in eine Datei (Unit) zu schreiben. Der Grund dafür ist, dass bei mehr als 60 bis 80 Zeilen Quellcode die Übersicht über die Programmstruktur verloren geht. Selbst die Unterteilung in Unterprogramme reicht ab einer bestimmten Größe von Programmen nicht mehr aus. SiSy erlaubt zwar in kleinen Programmen bzw. je Unit 10 Kilobyte Code. Das sind in Assembler zum Beispiel über 1000 Zeilen. Ein kleines Programm bzw. eine einzelne Unit sollte jedoch nie mehr als 80 bis 120 Zeilen haben. Wird diese Grenze erreicht, sollte das Programm in logische Einheiten (Units, Module) gegliedert werden. Dabei fasst man alles zusammen, was zu einer bestimmten Funktionalität oder einer bestimmten Baugruppe wie zum Beispiel dem AD-Wandler gehört. Physisch entstehen dabei mehrere Dateien, die für sich genommen wieder übersichtlich sind da diese dann nur 80 bis 120 Zeilen Code enthalten. Das Übersetzungsprogramm (Assembler, Compiler, Linker) sorgt dann dafür, dass alle einzelnen Units zu einem vollständigen Programm zusammengefügt werden.

Das folgende Kapitel erläutert an einem sehr kleinen Beispiel die Handhabung der Komponenten eines großen Programms, welches in mehrere Units zerlegt wird.

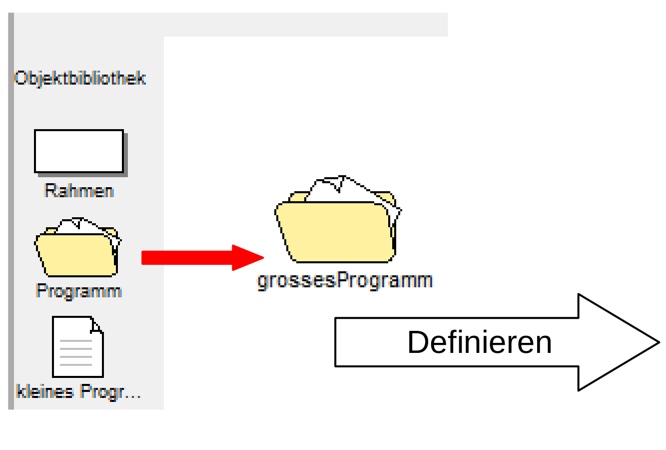
7.2 Vorbereitung

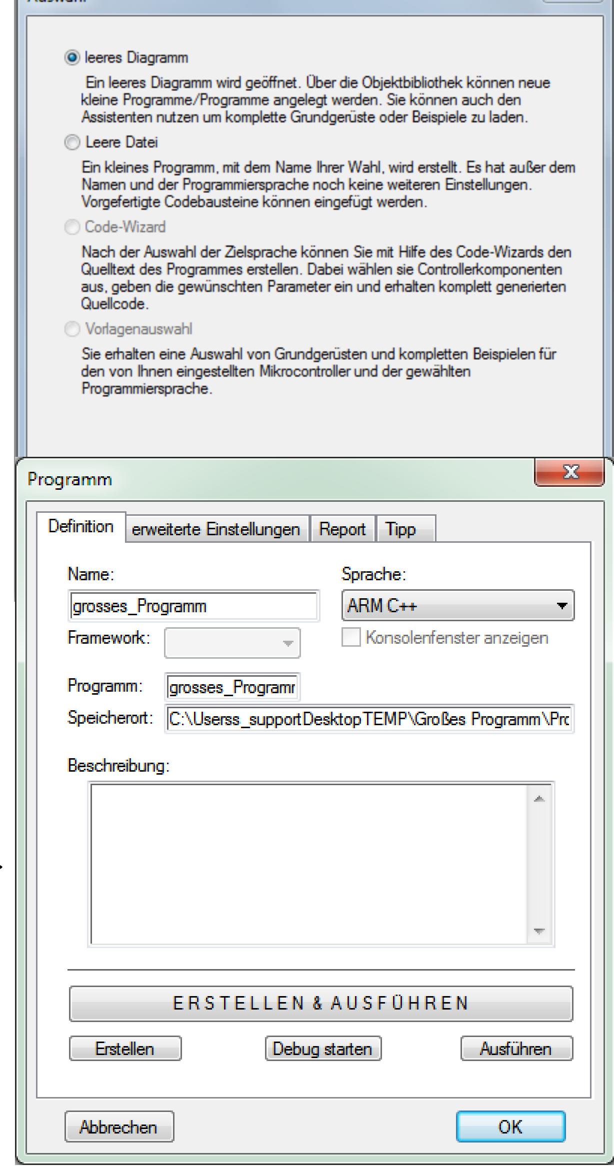
Starten Sie SiSy und legen Sie ein neues Projekt an. Wählen Sie das Vorgehensmodell „Programmierung“.

Nehmen Sie die Grundeinstellungen für die verwendete Hardware vor oder lassen Sie die Hardware automatisch suchen. Erstellen Sie ein leeres Diagramm!

Ziehen Sie aus der Objektbibliothek ein Objekt vom Typ „Programm“.

Legen Sie Name und Sprache für das Programm fest. Überprüfen Sie gegebenenfalls die Einstellungen unter „Extras AVR“.





7.3 Aufgabenstellung

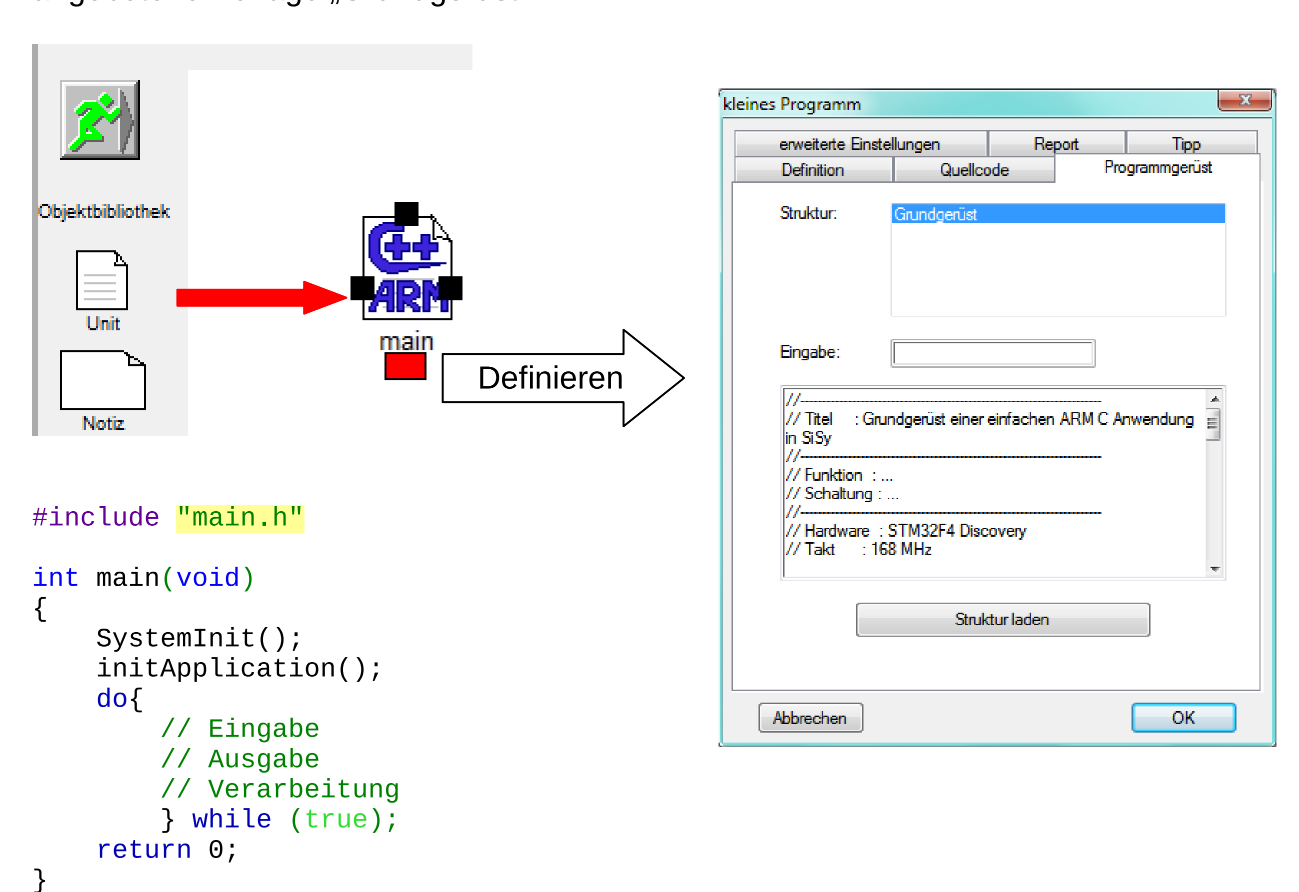
Es ist eine Mikrocontroller-Anwendung mit der Technik des großen Programms (Zerlegen in mehrere Units) zu entwerfen und in der Sprache C zu realisieren.

Aufgabe:
Entwickeln Sie eine Mikrocontrollerlösung, bei der ein Blinklicht erzeugt wird.

Schaltung:
GPIOD.13 = LED
GPIOD.12 = LED

7.4 Hauptprogramm erstellen

Öffnen Sie das Diagrammfenster für ein großes Programm, indem Sie auf dem Symbol rechte Maustaste -> Kontextmenü -> „Nach unten (öffnen)“ wählen. Legen Sie eine Unit an. Diese Unit bildet das Hauptprogramm. Nennen Sie die Unit „main“. Damit wird durch die Entwicklungsumgebung erkannt, dass es sich hierbei um das Hauptmodul handelt. Erstellen Sie hier das Hauptprogramm, nutzen Sie die angebotene Vorlage „Grundgerüst“.

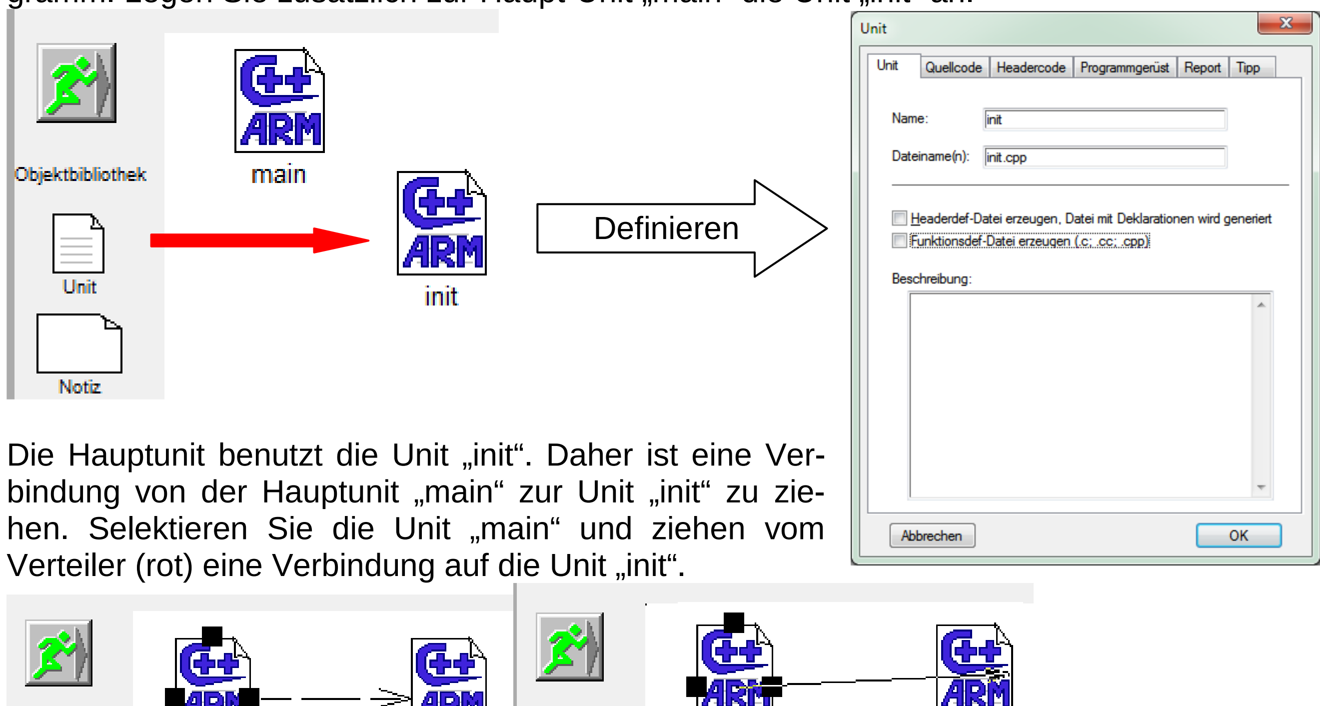



```
#include "main.h"

int main(void)
{
    SystemInit();
    initApplication();
    do{
        // Eingabe
        // Ausgabe
        // Verarbeitung
        } while (true);
    return 0;
}
```


7.5 Units (Unterprogramme) anlegen und verknüpfen

Zur Gliederung des großen Programms wird dieses in mehrere kleine Einheiten (Units/ Module) zerlegt. Diese Einheiten werden nach fachlichen/inhaltlichen Gesichtspunkten gebildet. So kann man alle Initialisierungsaufgaben in der Unit „init" zusammenfassen. Eine Unit kann aus einer Funktion/Unterprogramm oder mehreren Funktionen/Unterprogrammen bestehen. Im einfachsten Fall enthält jede Unit ein Unterprogramm. Legen Sie zusätzlich zur Haupt-Unit „main" die Unit „init" an.



Die Hauptunit benutzt die Unit „init". Daher ist eine Verbindung von der Hauptunit „main" zur Unit „init" zu ziehen. Selektieren Sie die Unit „main" und ziehen vom Verteiler (rot) eine Verbindung auf die Unit „init".

Dabei wird in der Hauptunit „main" ein Include-Eintrag für die Unit „init" erzeugt.
Erstellen Sie die Initialisierungsroutine für die benötigten digitalen Ein- und Ausgänge.

```
#include "init.h"

void initApplication()
{
    SysTick_Config(SystemCoreClock/100);
    // weitere Initialisierungen durchführen
    /* GPIOG Periph clock enable */
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);
    /* Configure PG6 and PG8 in output pushpull mode */
    GPIO_InitTypeDef  GPIO_InitStructure;
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_12 | GPIO_Pin_13;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIOD, &GPIO_InitStructure);
}
```

Ergänzen Sie den Code des Hauptprogramms.

```
#include "main.h"

int main(void)
{
    SystemInit();
    initApplication();
    GPIO_SetBits(GPIOD,GPIO_Pin_13);
    do{
        // Eingabe
        // Ausgabe
        // Verarbeitung
        GPIO_ToggleBits(GPIOD,GPIO_Pin_13|GPIO_Pin_12);
        delay(10000000);
        } while (true);
    return 0;
}
```

Ergänzen Sie nun die Units „sysTick“ und „delay“. Tragen Sie in diese folgenen Quell-code ein.

Für die „sysTick“ Unit

```
#include "sysTick.h"

extern "C" void SysTickFunction(void)
{
    // Application SysTick
    GPIO_ToggleBits(GPIOD,GPIO_Pin_13|GPIO_Pin_12);
}
```

Für die „delay“ Unit

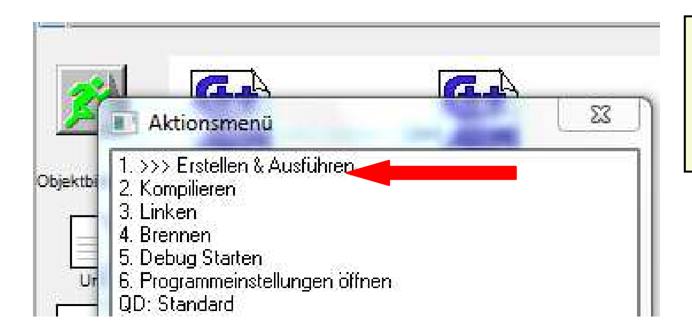
```
#include "delay.h"

void delay(vu32 nCount)
{
    while(nCount)
    {
        nCount--;
    }
}
```

Diese beiden Units sind noch mit der Hauptunit „main“ zu verbinden.

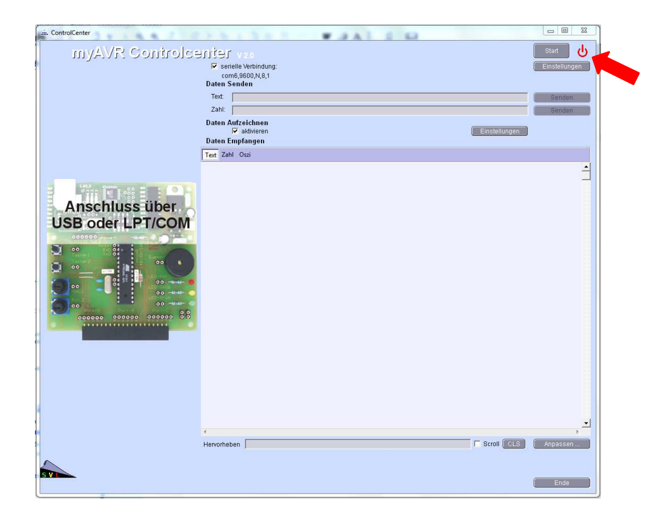
7.6 Übersetzen, Brennen und Test

Zum Übersetzen, Brennen und Testen wählen Sie im Aktionsmenü den entsprechenden Menüpunkt. Im Ausgabefenster erscheint das Protokoll der ausgeführten Aktionen. Des Weiteren öffnet sich die SiSy-Konsole.



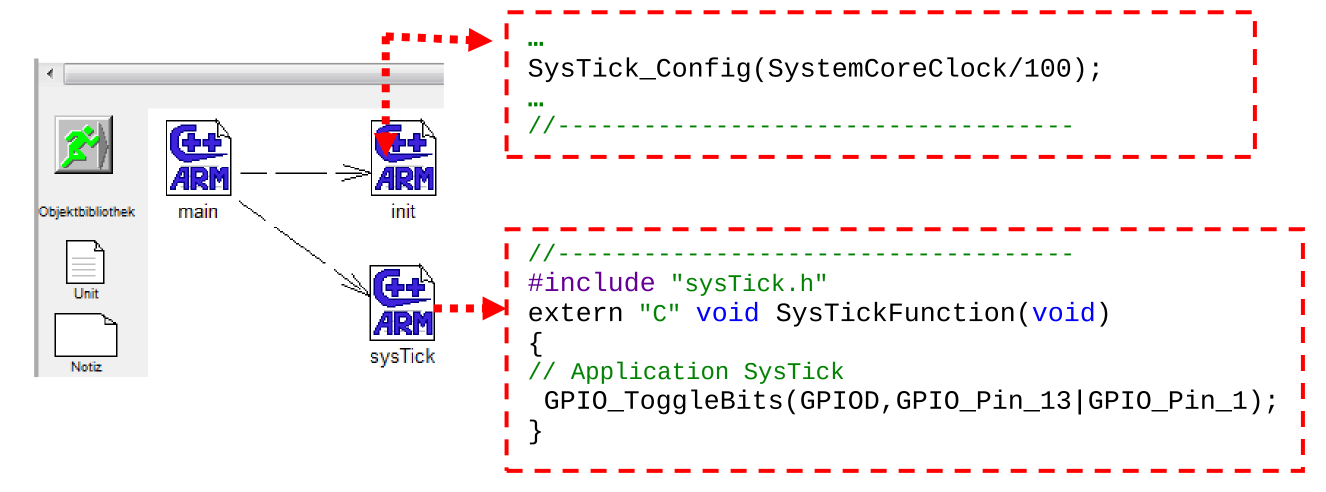
Kompiliere die Datei main.ccp.
Linke die Datei grossesProgramm.elf.
brenne Daten neu
Öffne myAVR ControlCenter
Ende.

Über das myAVR Controlcenter können Sie das myAVR Board Starten und Stoppen um die Mikrocontrollerlösung zu testen. Überprüfen Sie gegebenenfalls die Einstellungen entsprechend Absatz 6, „Das Mikrocontroller Controlcenter".



7.7 Interrupt-Service-Routine (ISR) im großen Programm

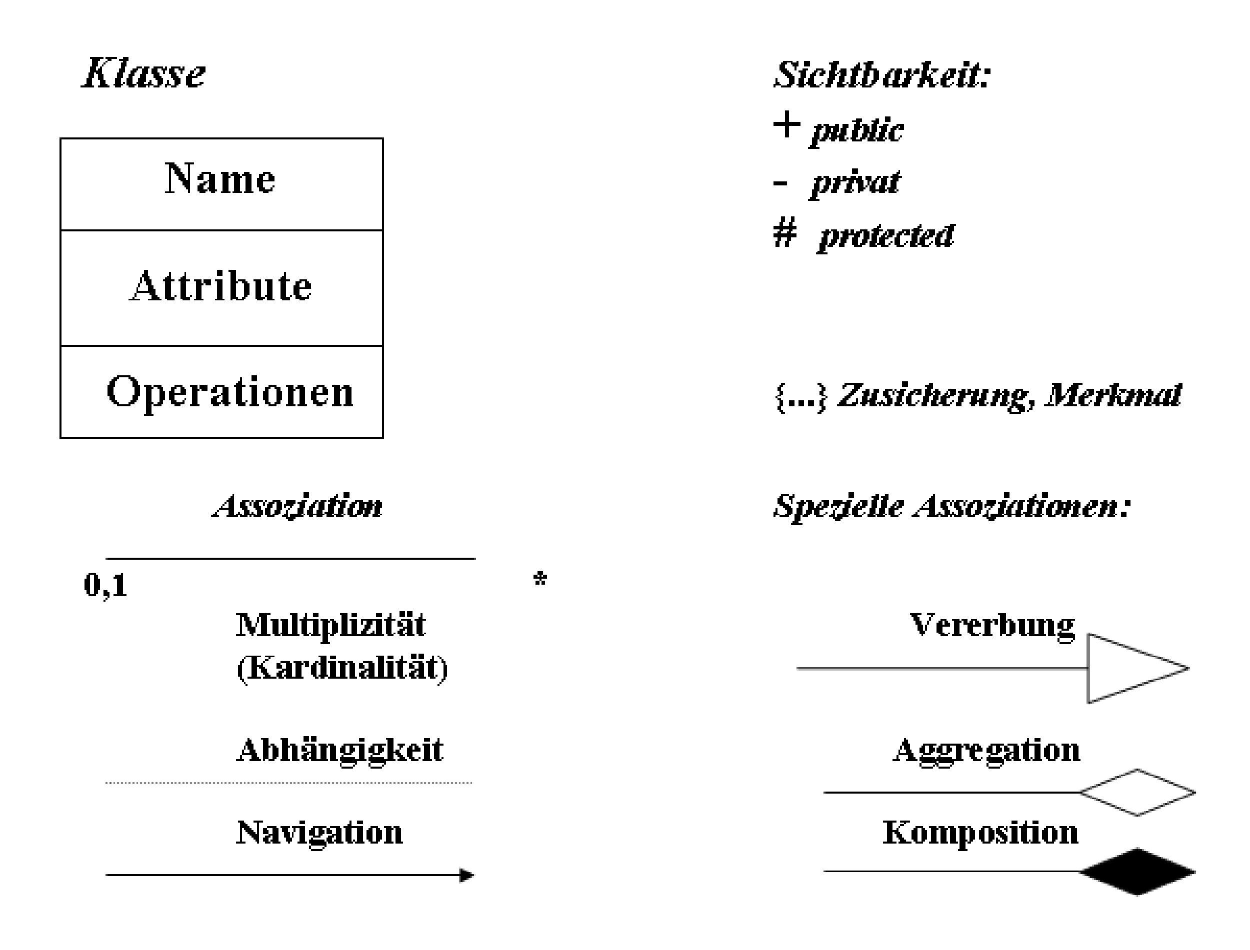
Interrupt-Service-Routinen (im weiteren ISR) sind besondere Formen von Unterprogrammen. Diese werden von einer Interruptquelle des Mikrocontrollers (Timer, ADC, UART, usw.) bei entsprechenden Ereignissen automatisch an beliebiger Stelle im Programmfluss aufgerufen (Unterbrechung, engl. Interrupt). Es ist nötig die Interruptquelle entsprechend zu konfigurieren. Es empfiehlt sich für jedes interruptfähige Gerät eine eigene Unit anzulegen. Diese ist mit der Haupt-Unit zu verbinden.



8 Entwicklung eines Klassendiagramms mit SiSy ARM

8.1 Einleitung

Mit objektorientierten Programmiersprachen hat der Entwickler mächtige Sprachmittel, um komplexe Systeme realisieren zu können. C++ ist eine weit verbreitete objektorientierte Standardsprache. Als Visualisierungsmittel objektorientierter Programme gilt die international standardisierte Beschreibungssprache UML (Unified Modeling Language).
SiSy ARM bietet dem Entwickler das UML Klassendiagramm mit Codegenerierung für ARM C++. Der folgende Abschnitt beschreibt die Handhabung des Klassendiagramms in SiSy ARM. Die Abbildung zeigt Ihnen eine Kurzübersicht der Modellierungselemente des UML Klassendiagramms.



Kurzübersicht Elemente des UML Klassendiagramms

Schreibweise von Attributen:
Attribute beginnen mit einem Kleinbuchstaben.

```
Sichtbarkeit name : Typ = Initialwert {Merkmal}
# temperatur : uint8_t = 25
```

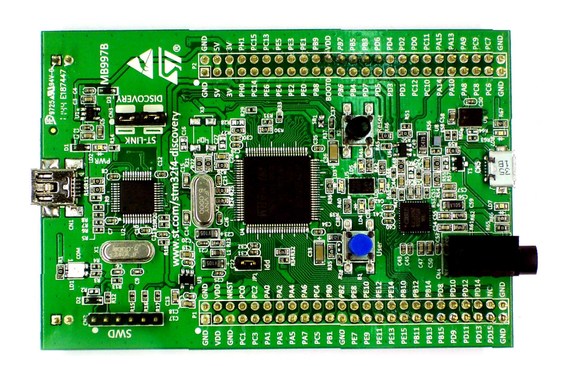
Schreibweise von Operationen:
Operationen beginnen mit einem Kleinbuchstaben.

```
Sichtbarkeit name (Parameter:Typ = Standardwert, ...) : Rückgabetyp {Merkmal}
+ setTemperatur ( temp : integer = 25 ) : bool
```


8.2 Aufgabenstellung

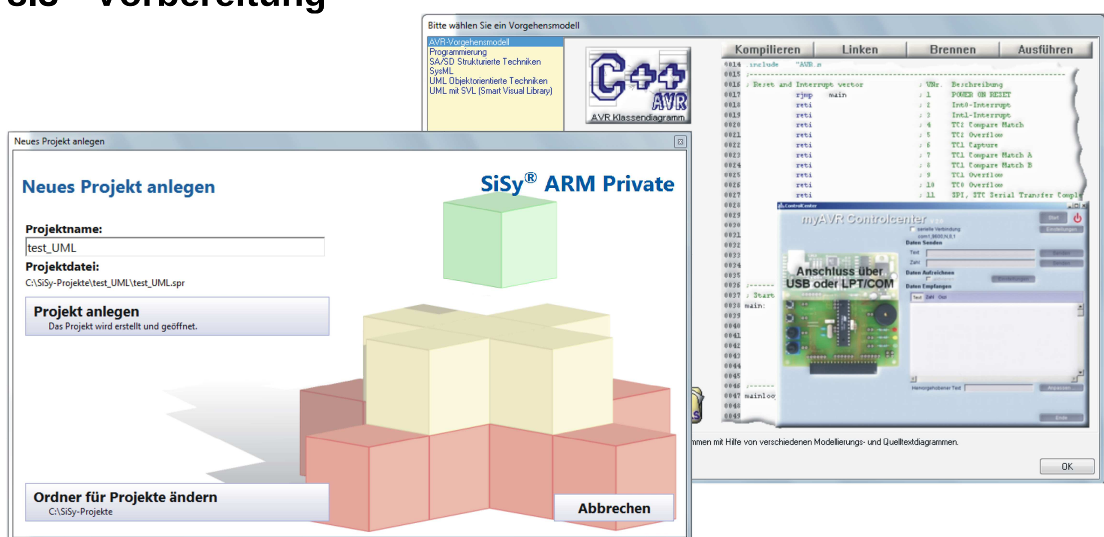
Um die Arbeitsweise und den Umgang mit dem Klassendiagramm in SiSy zu erlernen, realisieren Sie bitte das folgende einfache Beispiel.

Es soll eine Mikrocontrollerlösung entwickelt werden, bei der auf Tastendruck eine LED eingeschaltet wird. Dabei soll der Taster an Port GPIOA.0 und die LED an Port GPIOD.13 angeschlossen werden.



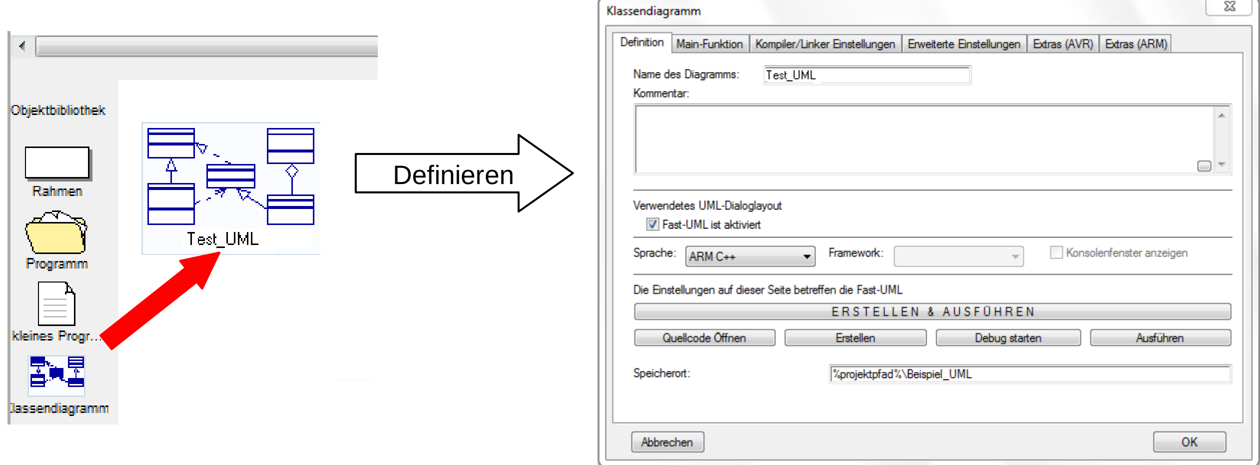
UML Beispiel

8.3 Vorbereitung



Starten Sie ggf. SiSy.

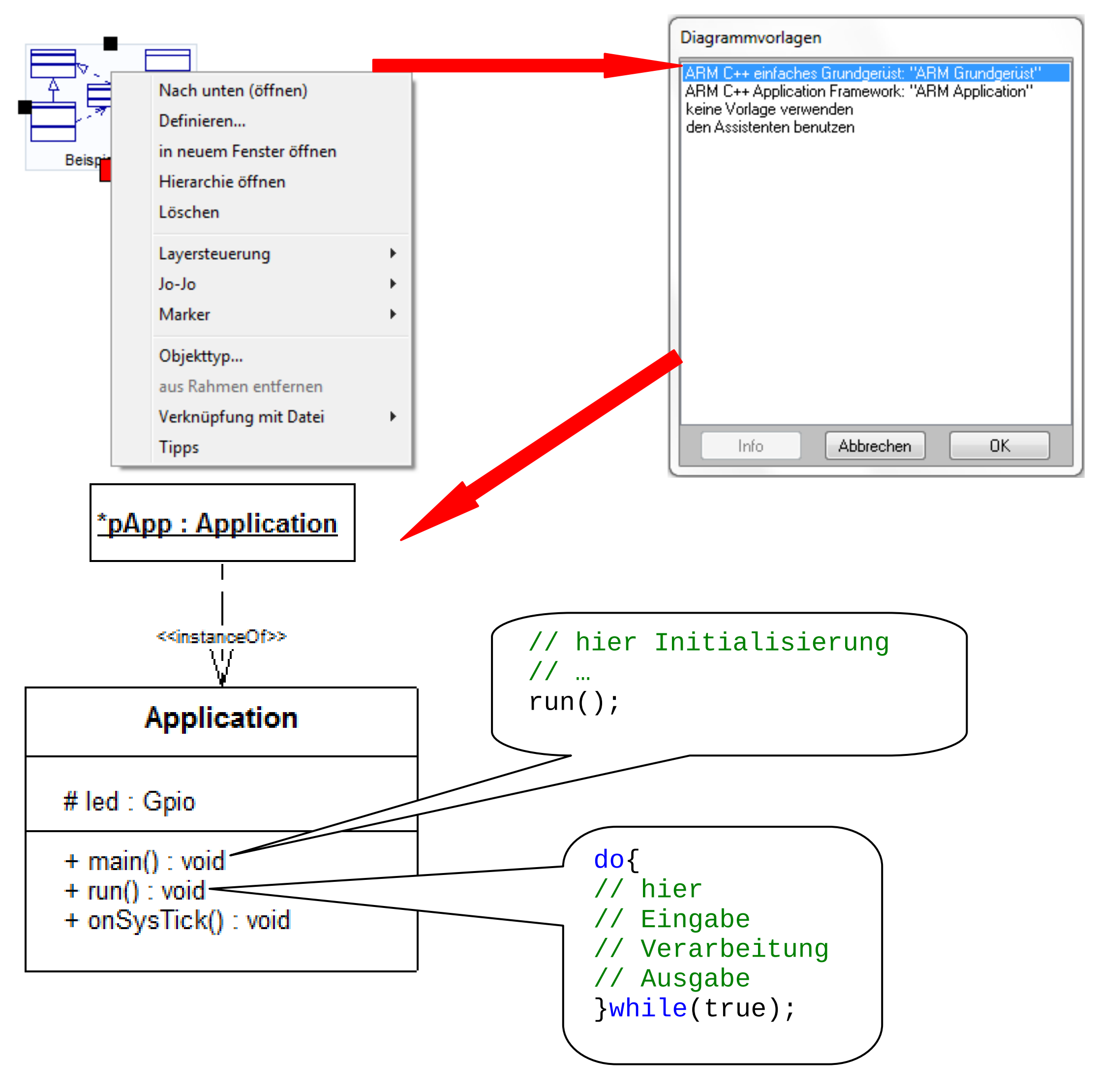
Erstellen Sie ein neues SiSy-Projekt (Hauptmenu -> Projekt -> Neu) mit dem Namen „Test_UML". Wählen Sie das Vorgehensmodell „ARM-Vorgehensmodell". Danach erscheint ein Dialogfenster mit möglichen Diagrammvorlagen.



Legen Sie im weiteren Verlauf die ARM-Grundeinstellungen für das Projekt fest. Ziehen Sie ein Objekt vom Typ „Klassendiagramm" per Drag & Drop in das Diagrammfenster. Legen Sie den Namen für das Klassendiagramm fest.

8.4 Grundstruktur laden

Um das Klassenmodell zu entwickeln, muss das Klassendiagramm geöffnet werden. Wählen Sie dazu auf dem Objekt rechte Maustaste -> Kontextmenü -> Nach unten (öffnen).



Die Struktur einer objektorientierten Mikrocontrollerlösung in SiSy erfordert im Klassenmodell eine Applikationsklasse (Hauptklasse), die sich dadurch auszeichnet, dass diese über eine Methode (Operation) mit dem Namen **`main`** verfügt. Der Codegenerator erzeugt eine Instanz dieser Klasse und ruft die Main-Methode auf.

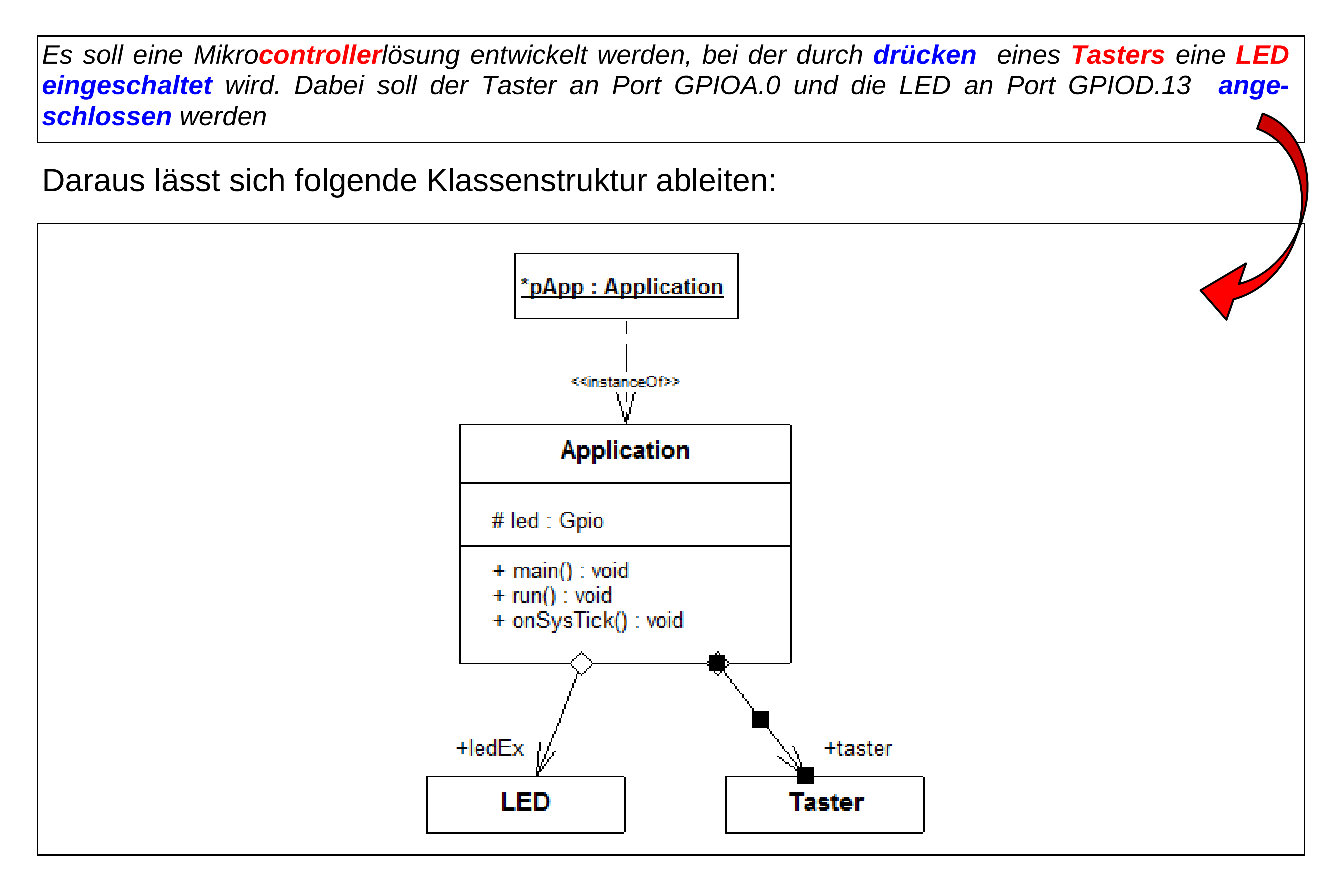
```
Systick::config(SystemCoreClock/100);
// Led initialisieren
Rcc rcc;
rcc.ahb1PeriphClockCmd(RCC_AHB1Periph_GPIOD,ENABLE);
led.initStruct.GPIO_Pin |= 1<<12;
led.initStruct.GPIO_Mode = GPIO_Mode_OUT;
led.initStruct.GPIO_Speed = GPIO_Speed_50MHz;
led.initStruct.GPIO_OType = GPIO_OType_PP;
led.initStruct.GPIO_PuPd = GPIO_PuPd_NOPULL;
led.init(GPIOD);
// Mainloop
this->run();
```

8.5 Systemstruktur entwerfen

Die Systemstruktur einer objektorientierten Anwendung bildet die Objekte und deren Beziehungen im Programm ab, welche im realen System als physische Elemente vorhanden sind. Als Bauplan der Objekte dienen Klassendeklarationen, welche die Eigenschaften (Attribute) und das Verhalten (Methoden/Operationen) der Objekte beschreiben. Das Klassendiagramm beschreibt also die Struktur der Klassen (Baupläne der Objekte) und die Beziehungen zwischen den Klassen. In unserer Aufgabenstellung finden wir die Objekte des Systems als **Substantive**, deren Beziehungen und Verhalten als **Verbalphrasen**.

*Es soll eine Mikro**controller**lösung entwickelt werden, bei der durch **drücken** eines **Tasters** eine **LED** **eingeschaltet** wird. Dabei soll der Taster an Port GPIOA.0 und die LED an Port GPIOD.13 **angeschlossen** werden*

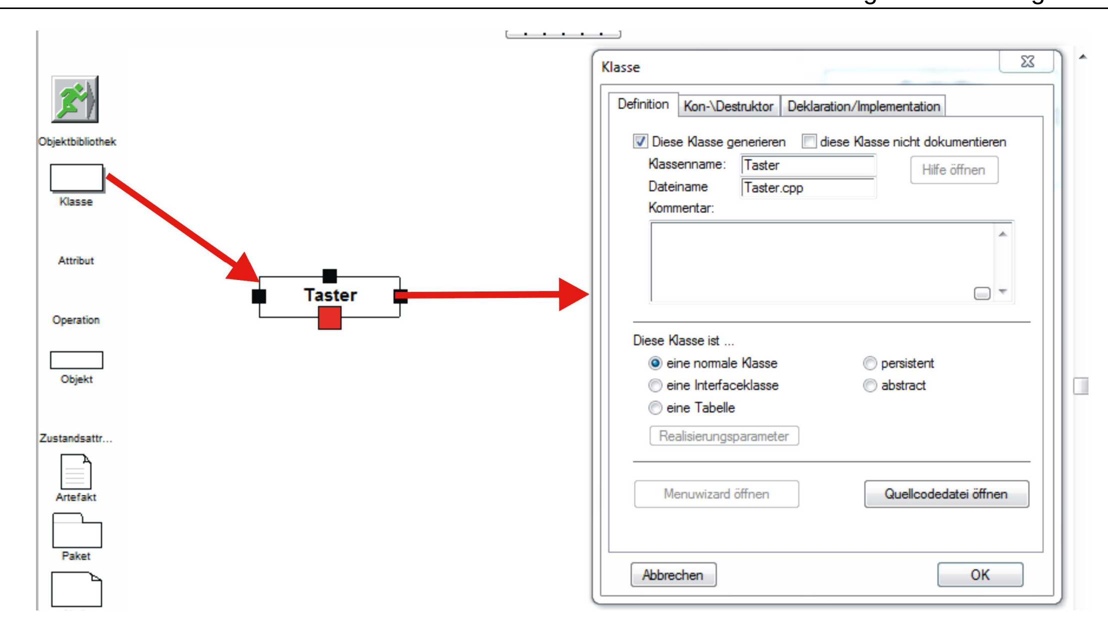
Daraus lässt sich folgende Klassenstruktur ableiten:



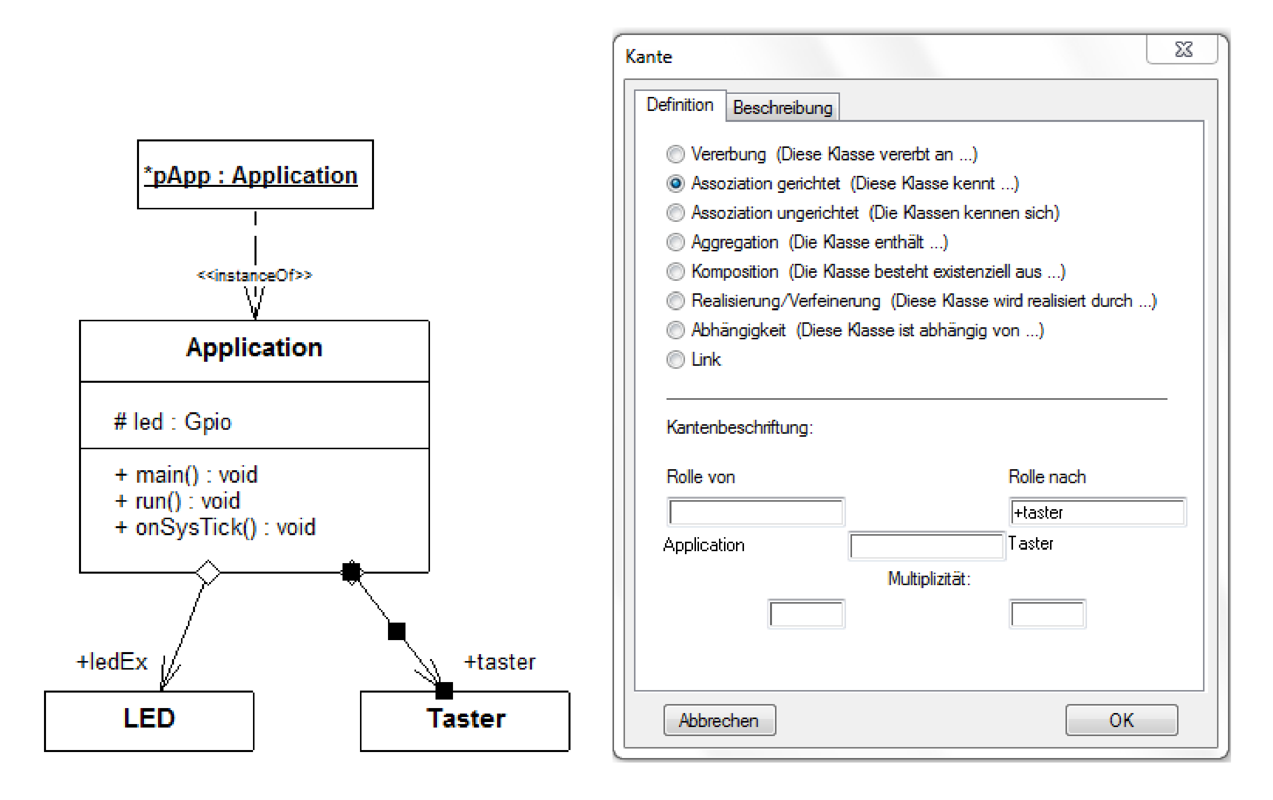
Klassenstruktur, Systementwurf mit dem UML Klassendiagramm

Zum Erstellen dieses Klassenmodells sind folgende Arbeitsschritte nötig:

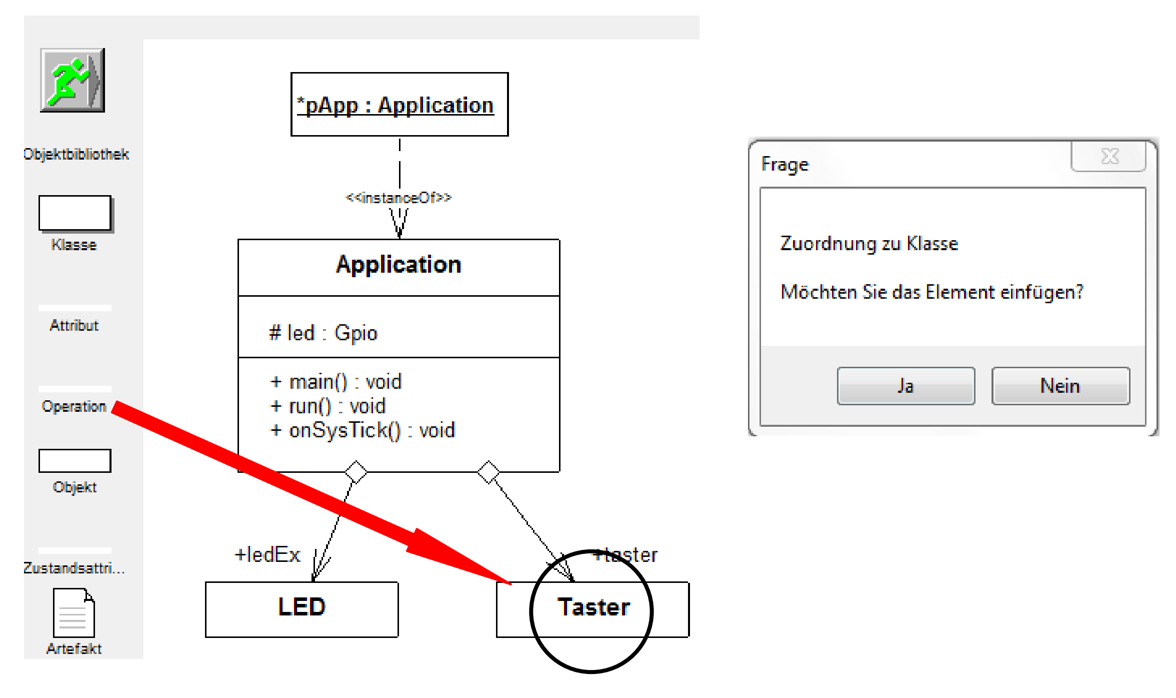
1. **Klassen einfügen und definieren**, ziehen Sie dazu das Objekt vom Typ „Klasse“ per Drag & Drop aus der Objektbibliothek in das Diagramm. Definieren Sie den Namen der Klasse und setzen die Option „diese Klasse generieren“.



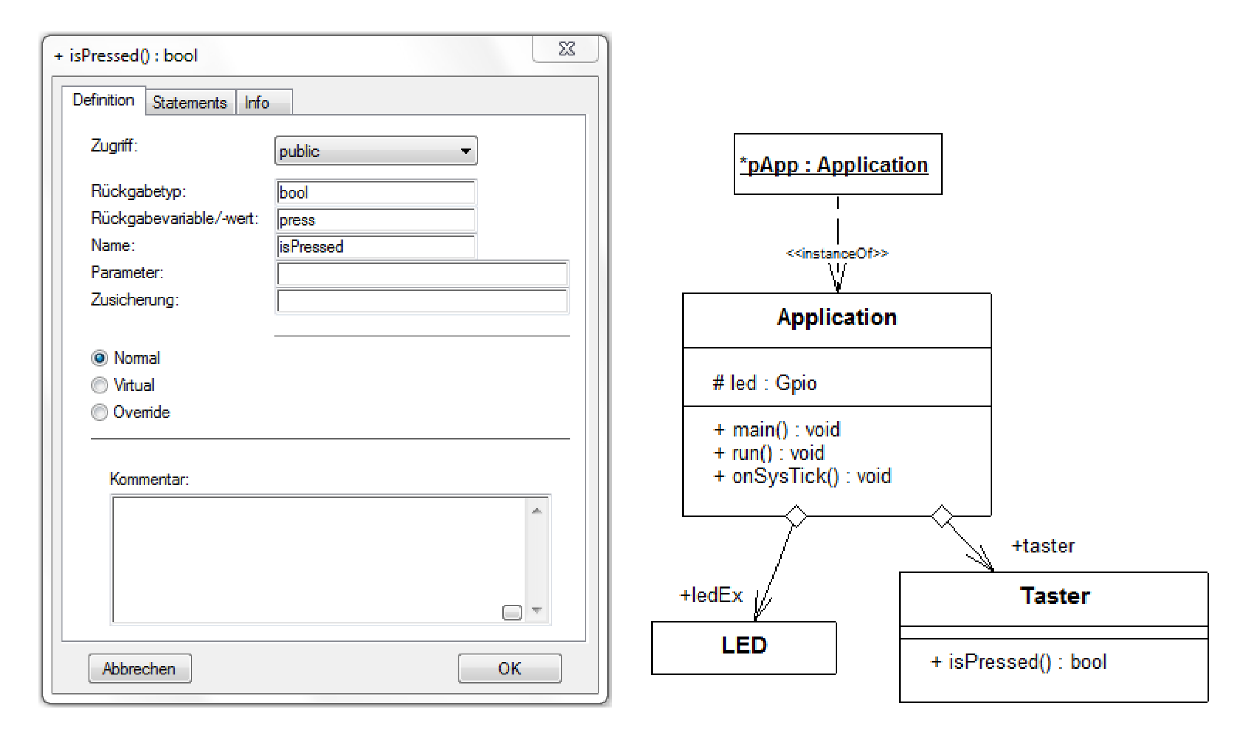
2. **Klassen verbinden**, selektieren Sie die Klasse, von der aus eine Verbindung gezogen werden soll. Ziehen Sie per Drag & Drop ausgehend vom roten Verteiler eine Verbindung auf die gewünschte Klasse. Wählen Sie den Verbindungstyp, zum Beispiel „Assoziation gerichtet“ und beschriften Sie die Verbindung.



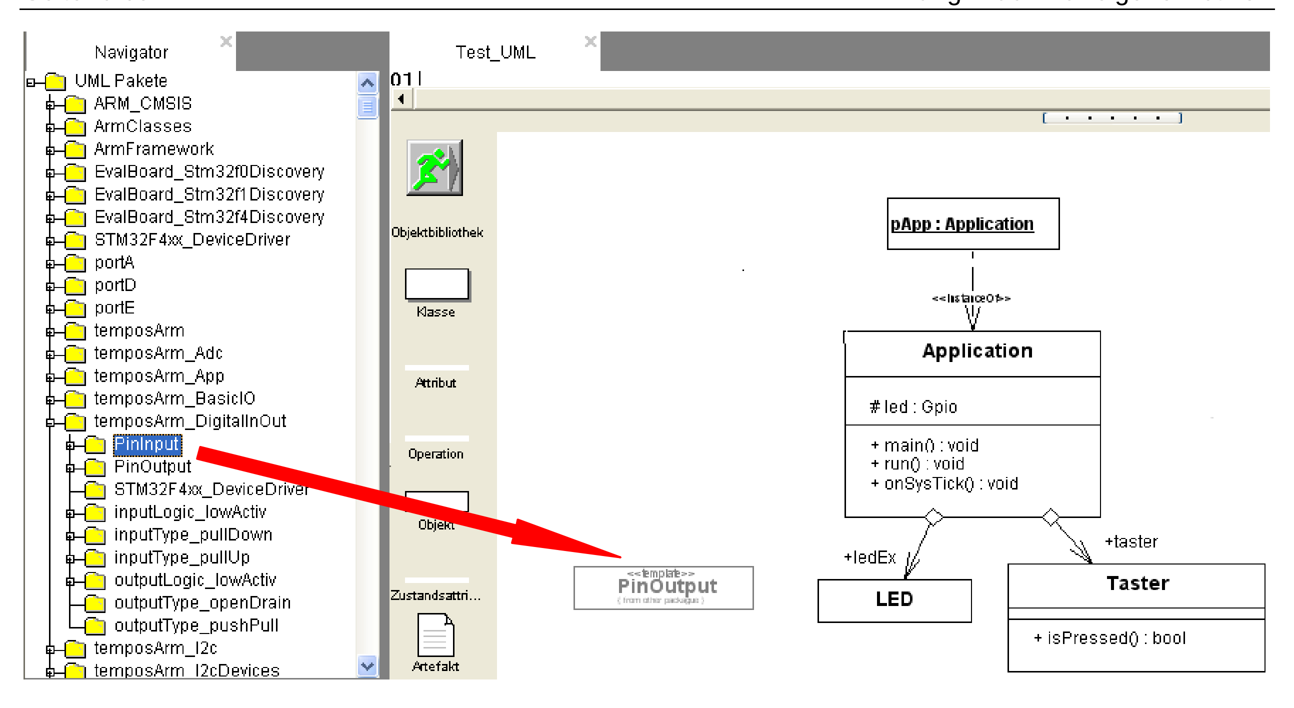
3. **Operationen und Attribute einfügen und definieren**, ziehen Sie dazu ein Objekt vom Typ „Operation“ oder „Attribut“ aus der Objektbibliothek auf die Klasse, in die das Attribut oder die Operation eingefügt werden soll. Bestätigen Sie die Sicherheitsabfrage zum Einfügen des Elementes.



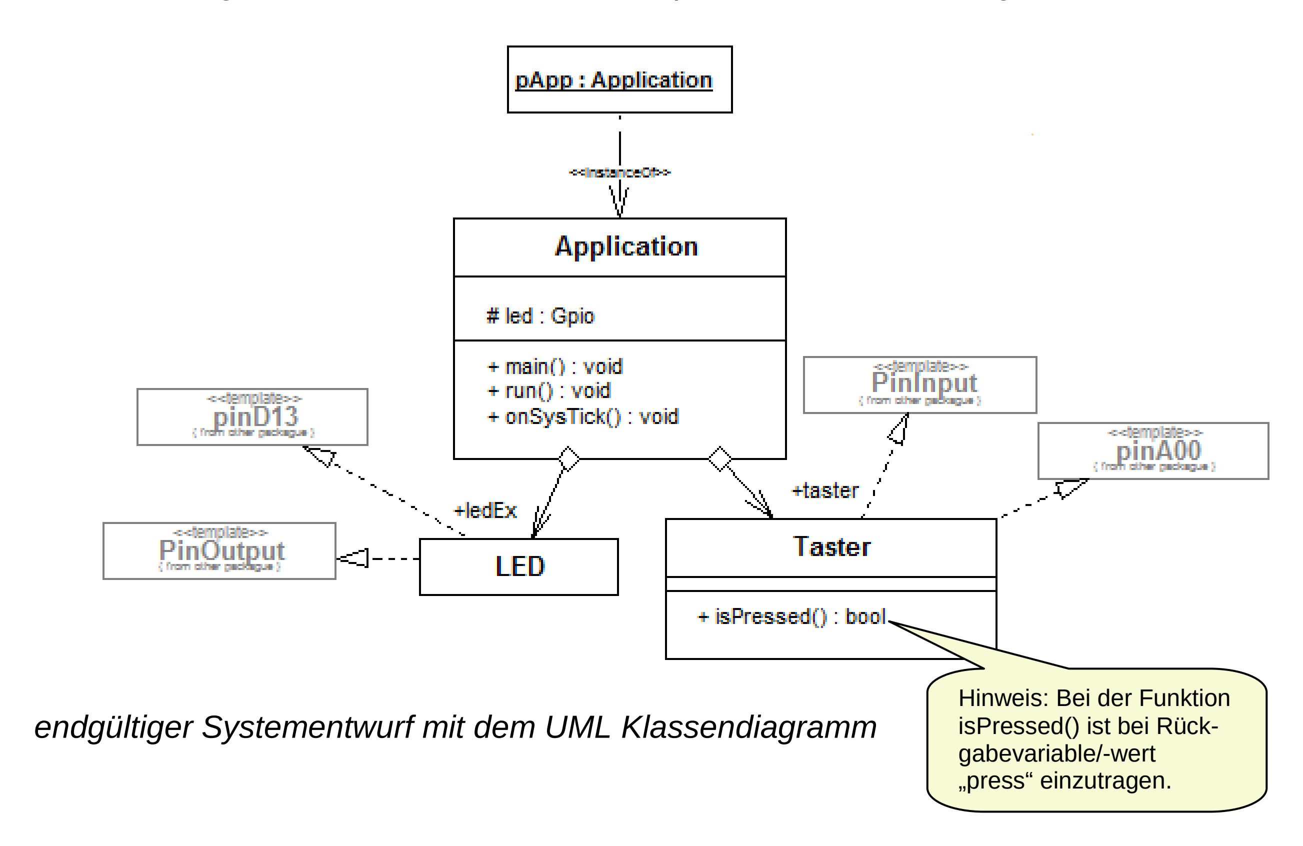
Definieren Sie Zugriff (Sichtbarkeit), Name, Typ und Parameter der Operation bzw. Zugriff (Sichtbarkeit), Name, Typ und Initialwert des Attributes.



4. **Templates einfügen**, klicken Sie mit der rechten Maustaste auf den „Navigator“ und wählen Sie die „UML-Pakete“ aus. Danach folgen Sie der Hierarchie und ziehen das gewünschte Template in das Diagramm. Zuletzt verbinden Sie das Template mit einer Klasse.



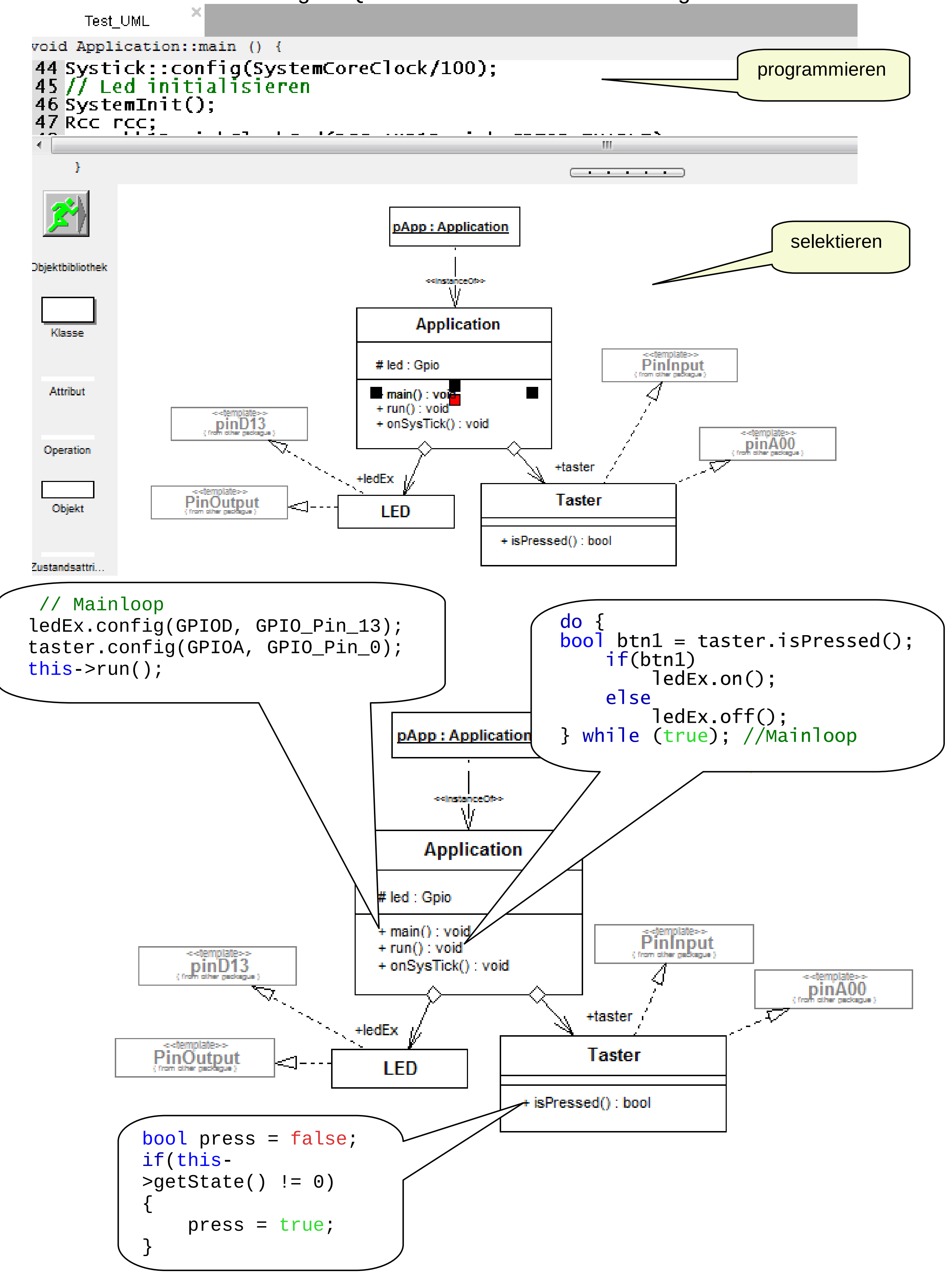
Vervollständigen Sie das Klassenmodell entsprechend der Abbildung.



endgültiger Systementwurf mit dem UML Klassendiagramm

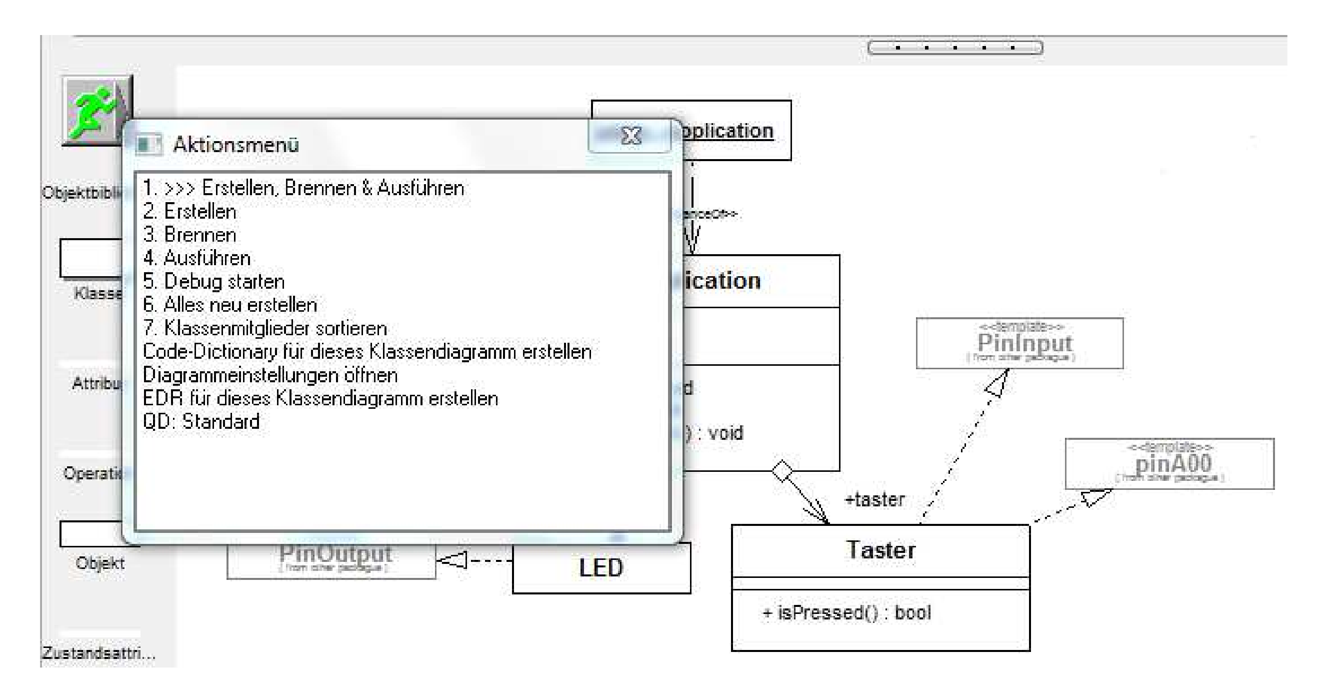
8.6 Systemverhalten programmieren

Die Operationen müssen mit der entsprechenden Logik versehen werden. Sie können den Quellcode der Operationen bearbeiten, indem Sie die gewünschte Operation selektieren und im Beschreibungs- / Quellcodefenster die Befehle eingeben.

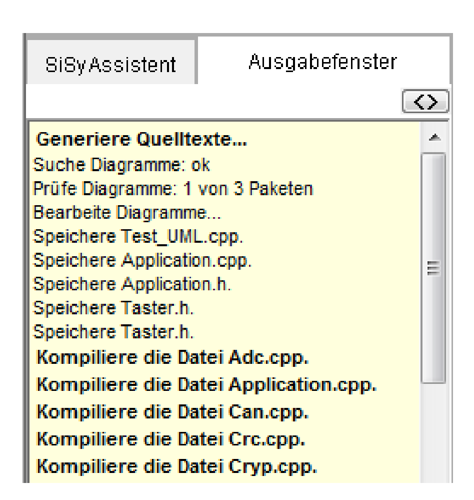


8.7 Übersetzen, Brennen und Testen

Die Codegenerierung aus dem UML Klassendiagramm, das Kompilieren, Linken und Brennen kann über das Aktionsmenü gestartet werden.



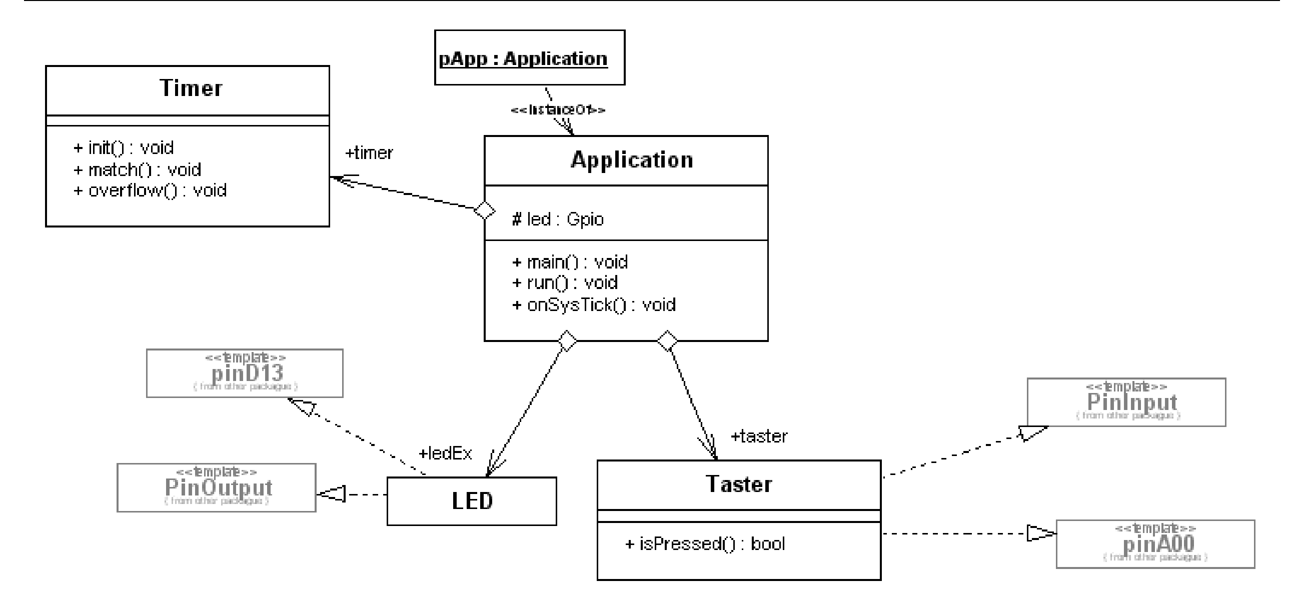
Sie erhalten im Ausgabefenster ein Protokoll der ausgeführten Aktionen.



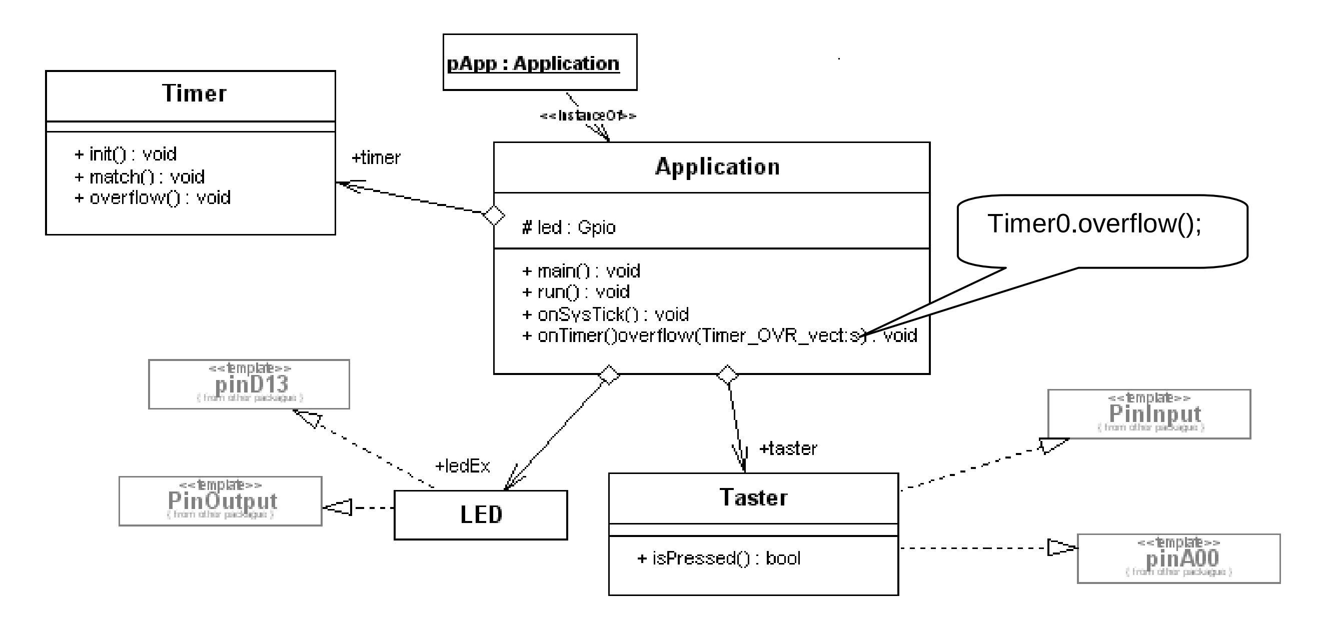
8.8 Interrupt-Service-Routinen (ISR) im Klassendiagramm

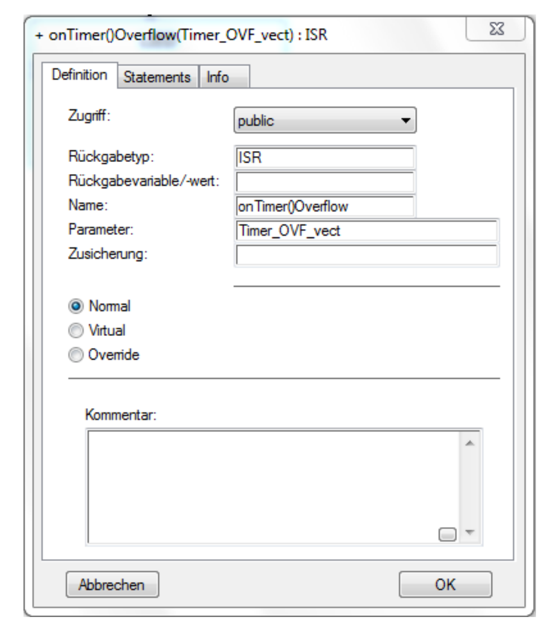
Interrupt-Service-Routinen (im weiteren ISR) werden in ARM C++ durch das Schlüsselwort ISR gekennzeichnet. Sie bilden eigenständige Funktionen. Die Besonderheit bei der Realisierung einer ISR liegt darin, dass es sich hier um ein C-Makro handelt und nicht um eine echte Funktion. Diese können also keine Methode einer Klasse sein. Um eine ISR im Klassendiagramm zu realisieren gehen Sie wie folgt vor (Erweiterung des Beispiels um die Klasse **`Timer`** und der **`ISR Timer0 Overflow`**):

1. Die Klasse für die interruptfähige Komponente modellieren und die Methode einfügen, welche beim Interrupt ausgeführt werden soll. In unserem Beispiel ist es die Klasse **`Timer`** mit der Methode **`overflow`**. Die Initialisierung des Timers muss entsprechend erfolgen.



2. Ergänzen Sie das Klassendiagramm weiter entsprechend der Abbildung.





Beispiel:

```
#include  <io.h>

// Instanz der interruptfähigen Klasse anlegen
Timer Timer0;

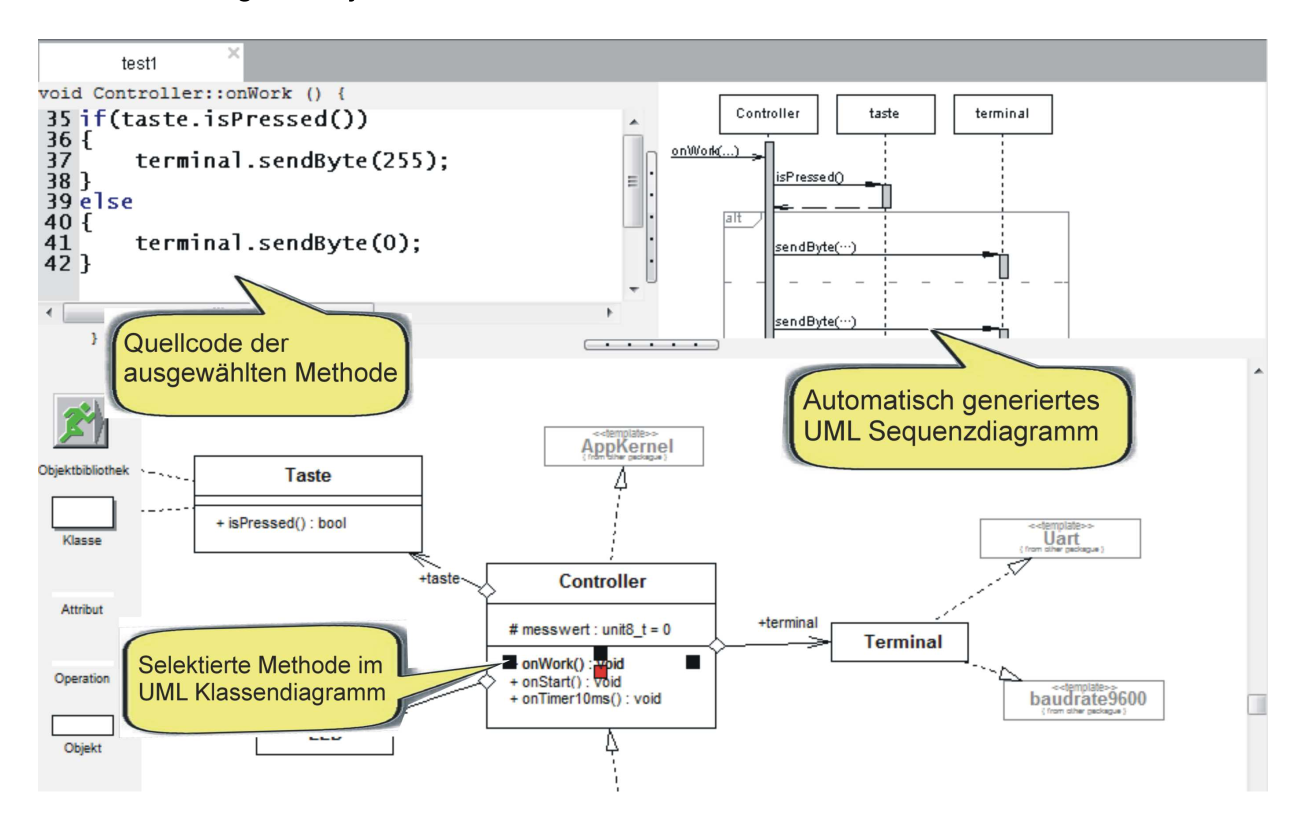
// Interuptmakro mit aufruf der betreffenden Methode
ISR (TIMER0_OVF_vect)
{
     Timer0.overflow();
}
```

In der Interrupt-Service-Routine rufen Sie einfach die Methode der betreffenden Klasse auf.

9 Der Sequenzdiagrammgenerator

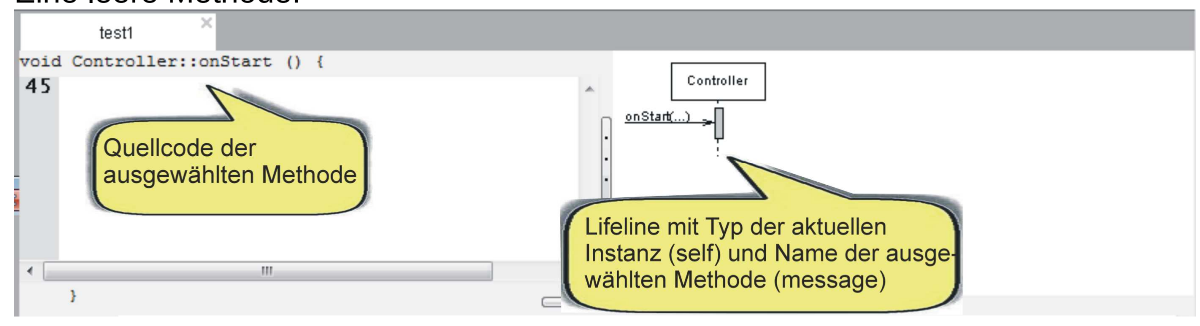
9.1 Einführung in das Sequenzdiagramm

Sequenzdiagramme sind eine wichtige Darstellungstechnik der UML für die Dokumentation des Systemverhaltens und der Interaktion von Objekten. SiSy verfügt über die Möglichkeit aus objektorientiertem Programmcode, wie zum Beispiel dem C++ Quellcode einer Klassenmethode, automatisch das entsprechende Sequenzdiagramm zu generieren. Dabei wird bereits während der Eingabe des Quellcodes simultan das Sequenzdiagramm erstellt. Der Entwickler ist damit in der Lage, die Reihenfolge der Nachrichten und die beteiligten Objekte mit deren Lebenszeit zu überblicken.

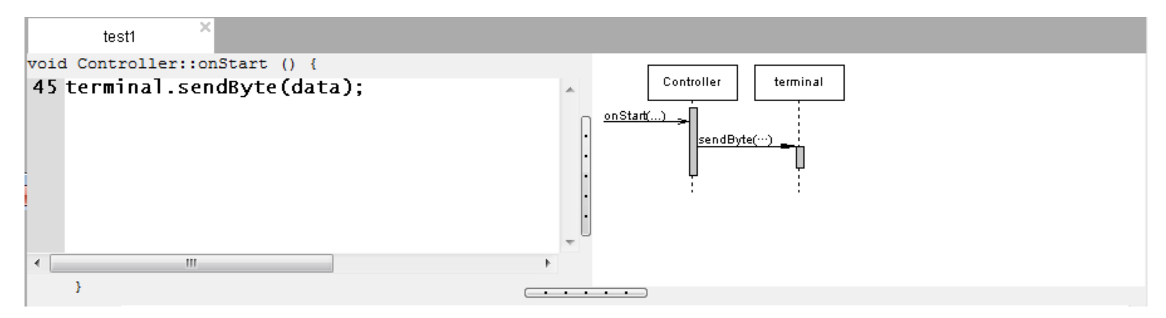


9.2 Sequenzen

Eine leere Methode:



Eine einfache synchrone Nachricht: *Instanz.Nachricht (Parameter);*



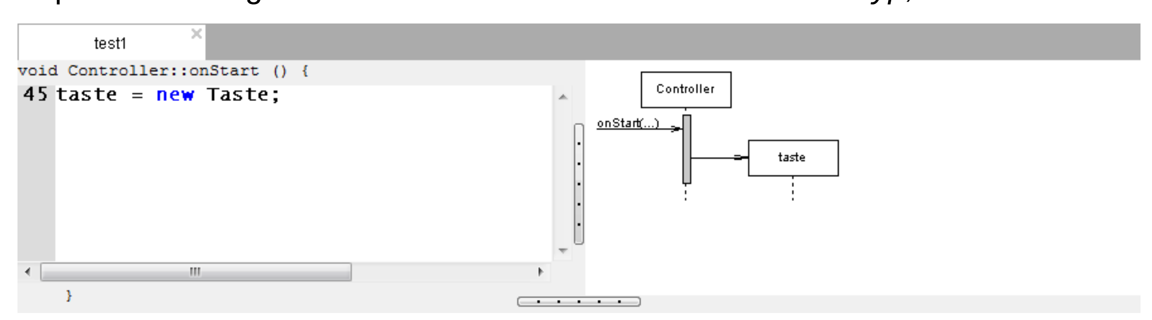
Eine Nachricht mit Antwort: *Variable = Instanz.Nachricht (Parameter);*

test1

```
void Controller::onStart () {
wert=Poti.getValue();
}
```

Controller
Poti
onStart(...)
getValue()
wert

Explizites Erzeugen einer Instanz mit new: *Instanz = new Typ;*



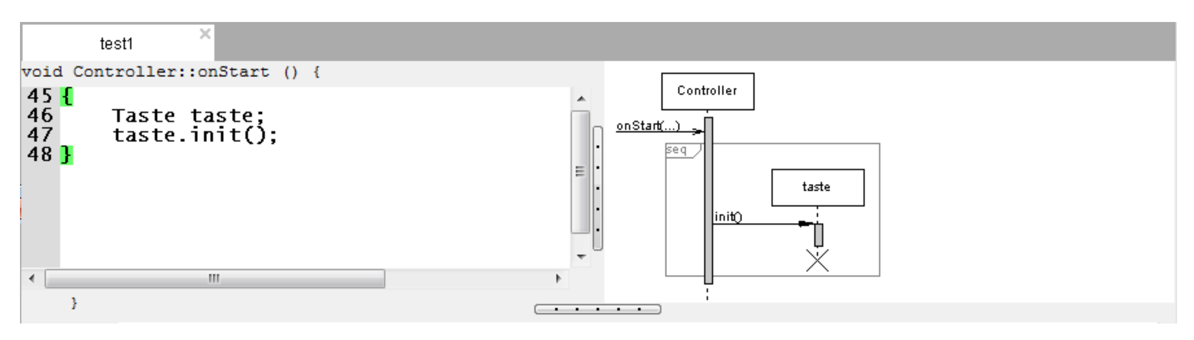
Explizites Zerstören einer Instanz mit delete: *delete Instanz;*

test1

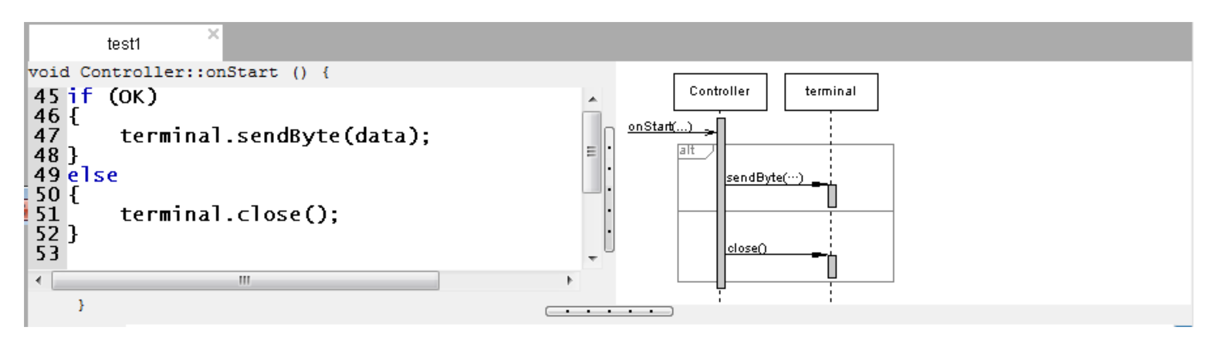
```
void Controller::onStart () {
taste = new Taste;
delete taste;
}
```

Controller
onStart(...)
taste

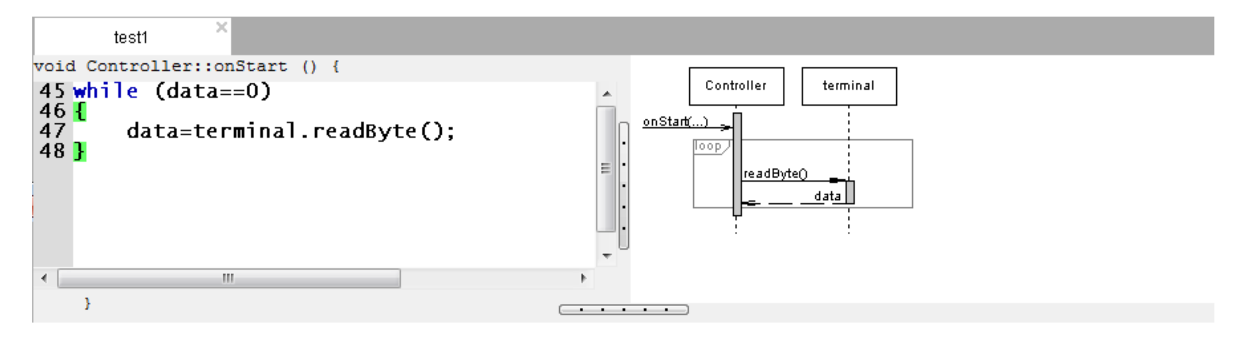
Eine lokale Instanz:



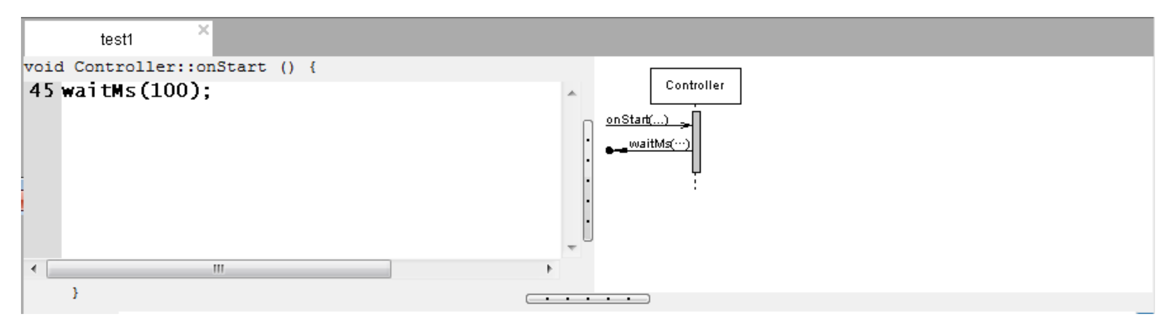
Eine Alternative:



Eine Schleife:



Nachricht an unbekannten Empfänger:

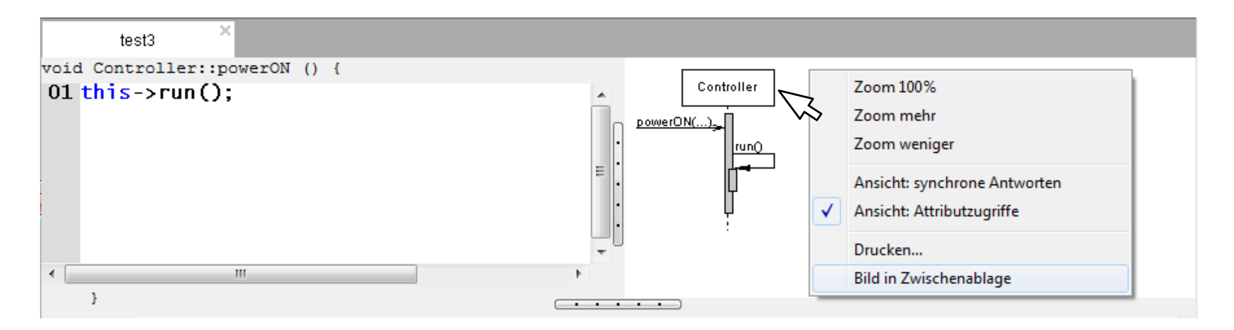


Nachricht an die aktuelle Instanz:

test3
void Controller::powerON () {
01 this->run();
}
Controller
powerON(...)
run()

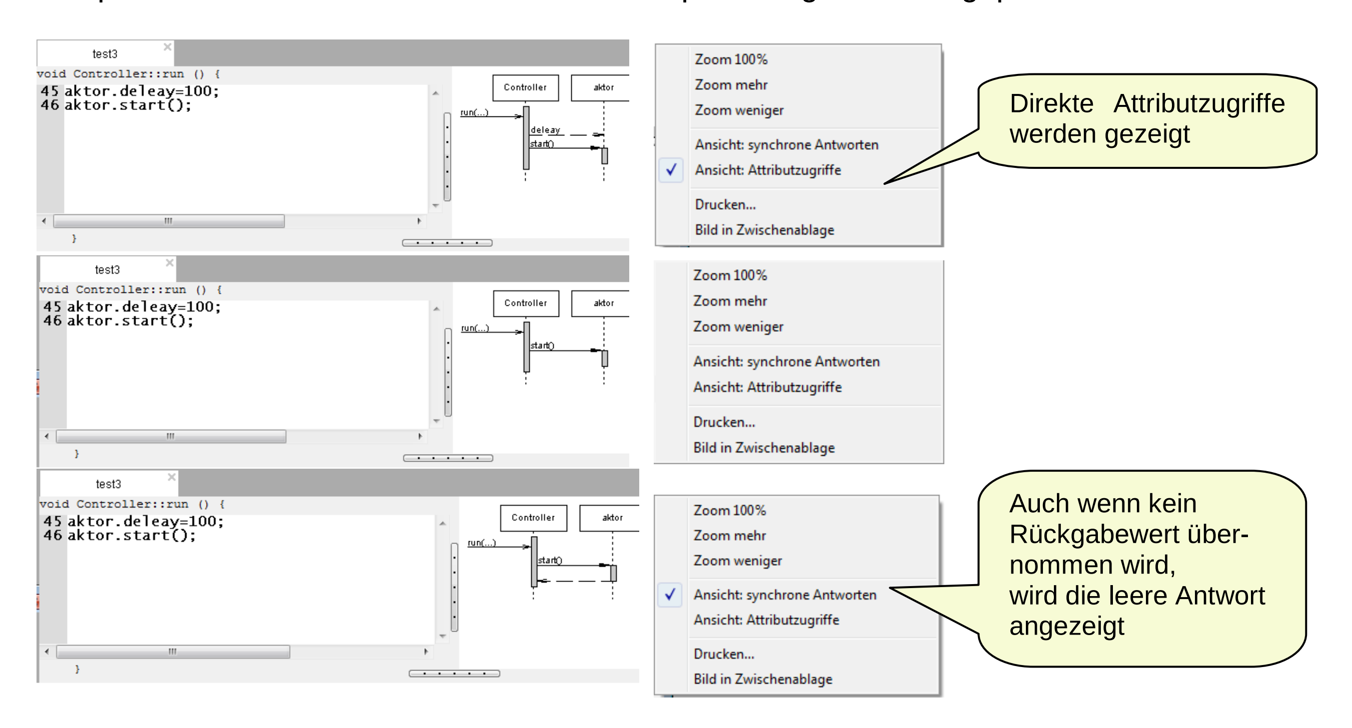
9.3 Sequenzen weiter verwenden

Sie können das automatisch erzeugte Sequenzdiagramm weiter verwenden, indem Sie es auf einen Drucker ausgeben oder die Darstellung in die Zwischenablage kopieren um diese zum Beispiel in eine Textverarbeitung einzufügen.



9.4 Besonderheiten des SiSy-Sequenzdiagramms

Für die direkten Zugriffe auf öffentliche Attribute einer Klasse trifft die UML derzeit keine verbindlichen Regeln. Es wird davon ausgegangen, dass Klassenattribute prinzipiell geschützt sind und nur über Methoden (Operationen) ein Zugriff erfolgen kann (Kapselung und Nachricht). Des Weiteren kann man davon ausgehen, dass auch eine leere (`void`) Antwort eine Antwort ist und angezeigt werden sollte. Dazu kann die Darstellung entsprechend über das Kontextmenü des Sequenzdiagramms angepasst werden.



10 Programmieren mit dem UML Zustandsdiagramm

10.1 Einführung in die Zustandsmodellierung

Viele Problemstellungen in der Softwareentwicklung lassen sich als eine Folge von Zuständen und Zustandsübergängen (Zustandsautomat, state machine) verstehen und auch als solche implementieren. Dabei handelt es sich im Quellcode um oft recht aufwendige Fallunterscheidungen bzw. if/else Konstruktionen. SiSy ist mit dem entsprechenden Add-On in der Lage, aus UML-Zustandsdiagrammen den entsprechenden Quellcode automatisch zu generieren.

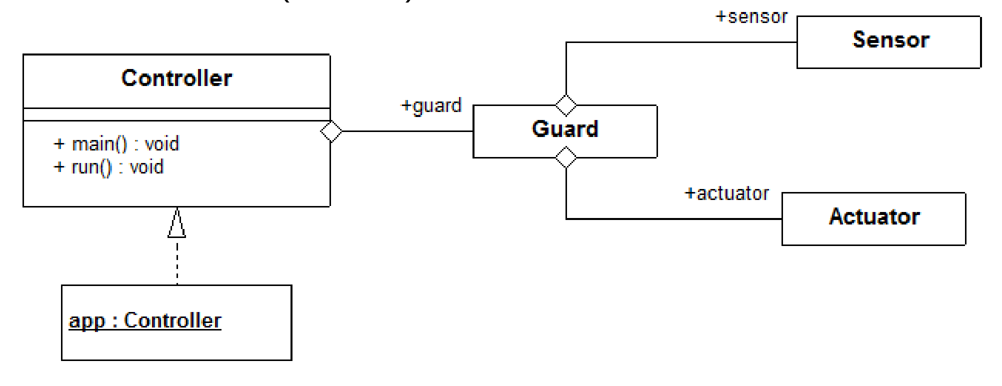
Notationsübersicht Zustandsdiagramm (Auszug)

Notation	Bezeichnung	Beschreibung
	Startknoten	Ein Startknoten aktiviert Abläufe. Startknoten besitzen nur ausgehende Kanten.
Zustand entry / wird beim Betreten ausgeführt do / wird wiederholt während des Zusandes ausgeführt event / wird bei einem bestimmten Ereignis ausgeführt exit / wird beim Verlassen ausgeführt	Zustand	Zustände eines Objektes sind gekennzeichnet durch zustandsspezifische Aktivitäten beim Einnehmen, während und beim Verlassen des Zustandes (entry, do, exit)
	Zustandsübergang	Ein Zustandsübergang (transition) repräsentiert mindestens zwei Aktivitäten. Eine Aktivität vom Typ „exit“ beim alten und eine Aktivität vom Typ „entry“ beim neuen Zustand.
	Endknoten	Ein Endknoten beendet Abläufe und besitzt nur eingehende Kanten.

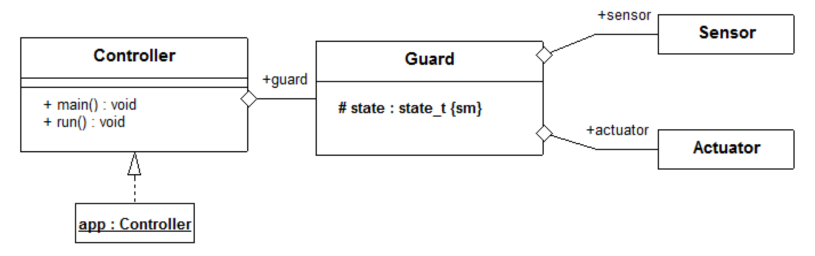
10.2 Erstellen von Zustandsdiagrammen

Das Zustandsdiagramm mit Quellcodegenerierung erhalten Sie über ein spezielles Attribut im Klassendiagramm der UML. Dieses Zustandsattribut kann mit einem Zustandsdiagramm verfeinert werden. Dabei wird der Quellcode in spezielle Klassenmethoden generiert. Der Entwickler kann selbst Methoden definieren, die an der Verarbeitungslogik des Zustandsautomaten (state machine) beteiligt sind. Voraussetzung ist es, ein neues Projekt mit einem Klassendiagramm anzulegen.

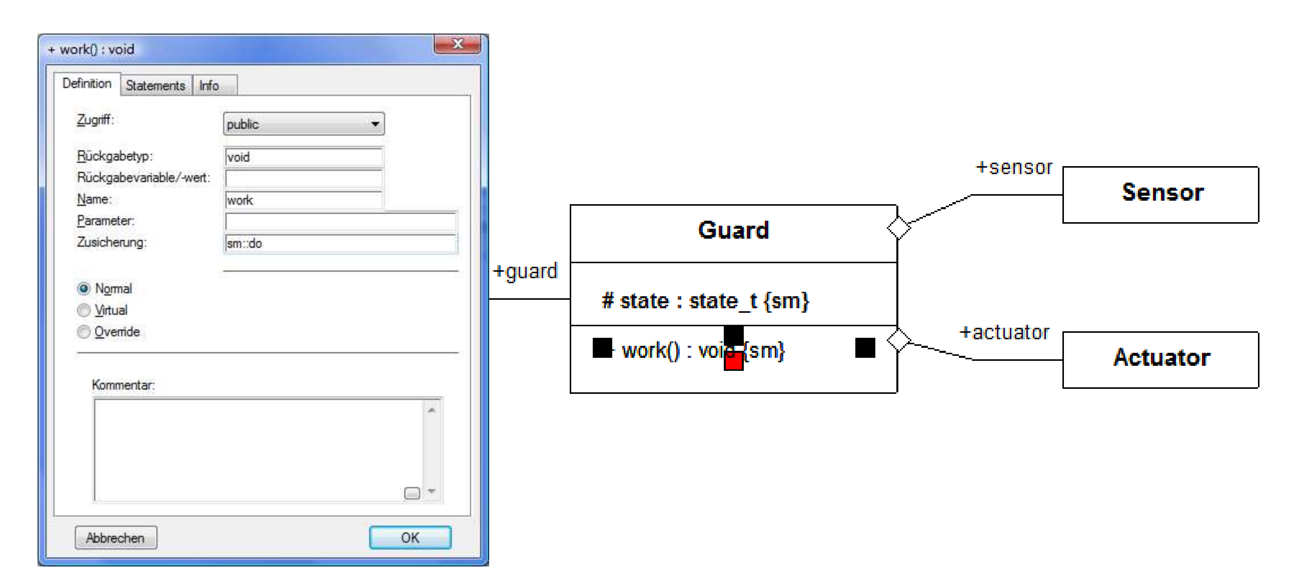
Die Aufgabe welche mit einem Zustandsdiagramm gelöst werden soll lautet wie folgt: Ein Controller steuert einen Wächter (guard) der einen Sensor überwacht und entsprechend einen Aktor (actuator) ansteuert.



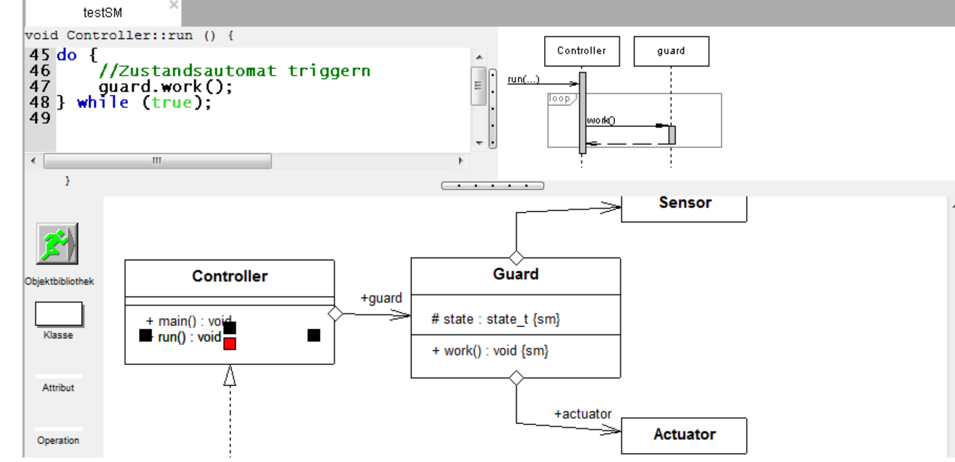
Der Wächter (guard) ist ein Zustandsautomat (state machine). Somit ist der Klasse „Guard“ ein Zustandsattribut zuzuordnen. Dazu muss ein Objekt vom Typ *Zustandsattribut* per Drag und Drop aus der Objektbibliothek in die betreffende Klasse gezogen werden. Die Zugehörigkeit zum Zustandsdiagramm wird mit der Zusicherung {sm} gekennzeichnet.



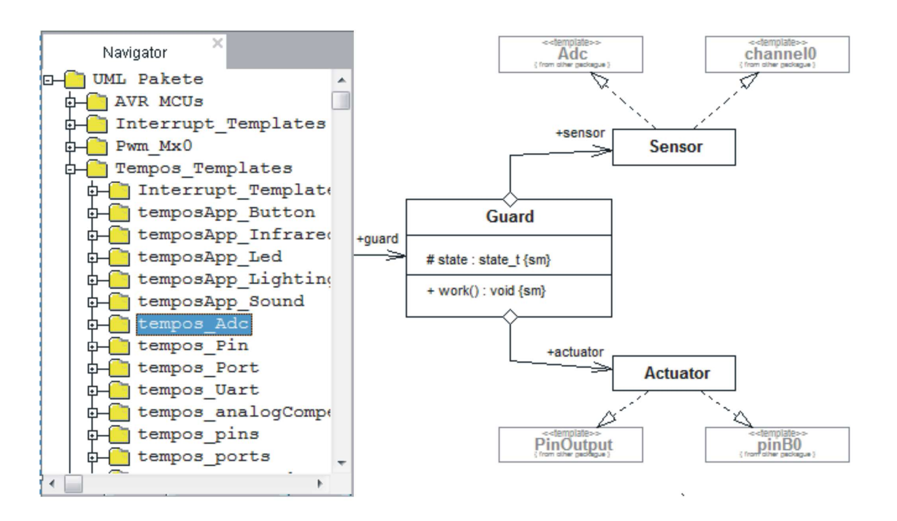
Für die Verarbeitungslogik des Zustandsautomaten soll eine eigene Methode angelegt werden, welche in der Hauptschleife der Mikrocontrolleranwendung (mainloop) zyklisch aufgerufen wird. Damit die Methode im Zustandsdiagramm verfügbar ist, muss diese mit einer entsprechenden Zusicherung versehen werden (z.B.: {sm::do}). Dazu muss aus der Objektbibliothek per Drag und Drop eine Operation (Methode) in die betreffende Klasse gezogen und wie folgend dargestellt definiert werden.



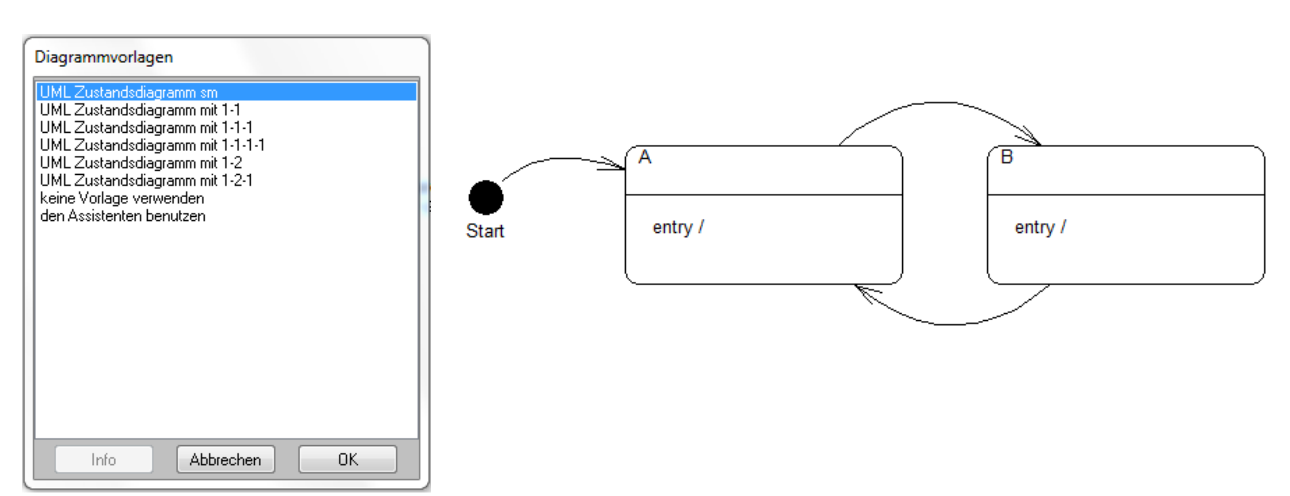
Diese Methode benötigt keine Statements. Der Codegenerator fügt beim Bilden der Anwendung hier Code entsprechend der Modellierung im Zustandsdiagramm ein. Es ist jedoch nötig, die Verarbeitungsfunktion kontinuierlich aufzurufen um die Zustandswechsel auszulösen (trigger). Das erfolgt in diesem Fall in der Methode *run*(). Es gibt in den SiSy-Klassenbibliotheken spezielle Klassen, die bereits einen Zustandsautomaten abbilden. Wird von solchen Basisklassen abgeleitet, sind die Auslöser (trigger) für den Zustandsautomaten bereits implementiert.



Für eine schnelle Realisierung der Klassen „Sensor“ und „Actuator“ wird im folgenden Beispiel das myAVR Programmiermodell *Tempos* benutzt. Dabei müssen den Klassen lediglich geeignete Templates (Muster) aus der Tempos-Bibliothek zugewiesen werden.

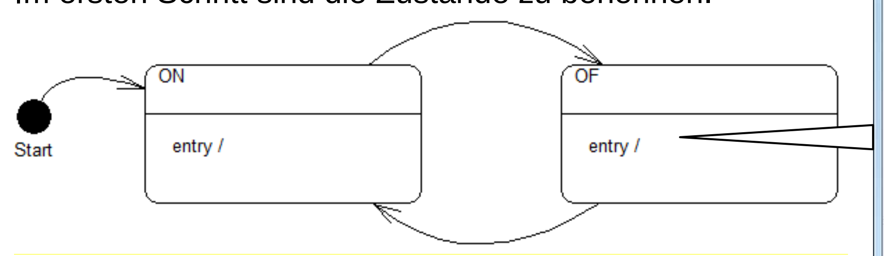


Die Modellierung der Zustände und Zustandswechsel erfolgt in Verfeinerung des Zustandsattributes. Dazu selektieren Sie das Zustandsattribut und wählen im Kontextmenü den Menüpunkt *Nach unten (öffnen)*. Dabei wird bereits eine Auswahl typischer Zustandsvarianten als Diagrammvorlage angeboten. Wählen Sie die Vorlage 1-1.

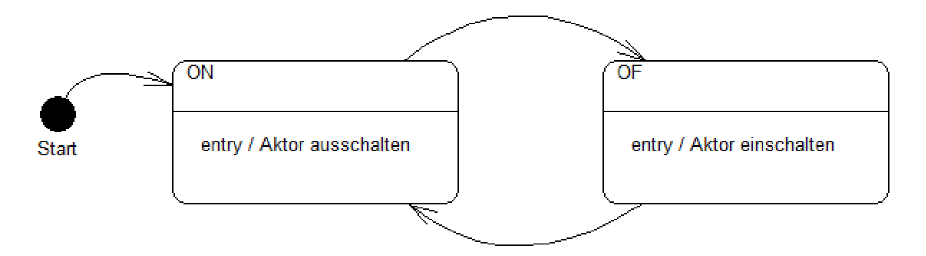


Die Zustände des Wächters sollen ON und OFF sein. Dabei wird nach dem Start der Zustand OFF eingenommen. Die Zustandswechsel erfolgen in Abhängigkeit der Sensorwerte. Wird der Zustand OFF eingenommen, muss der Aktor ausgeschaltet werden, wird der Zustand ON eingenommen, ist der Aktor einzuschalten. Während der Zustände ist der Sensor fortlaufend zu überwachen.

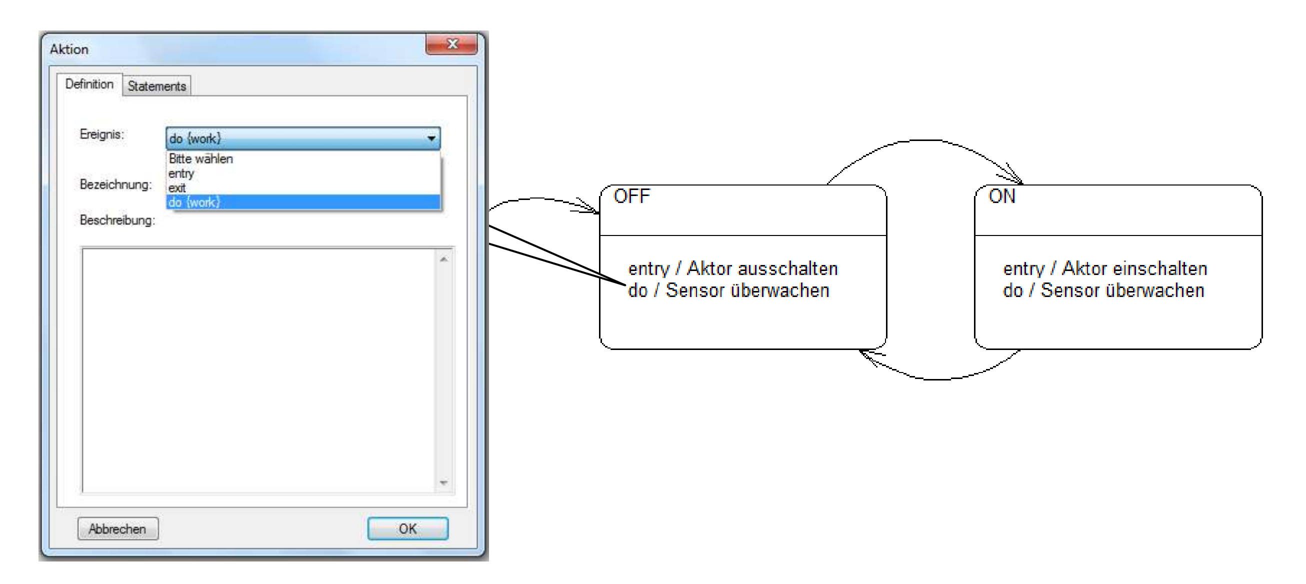
Im ersten Schritt sind die Zustände zu benennen.



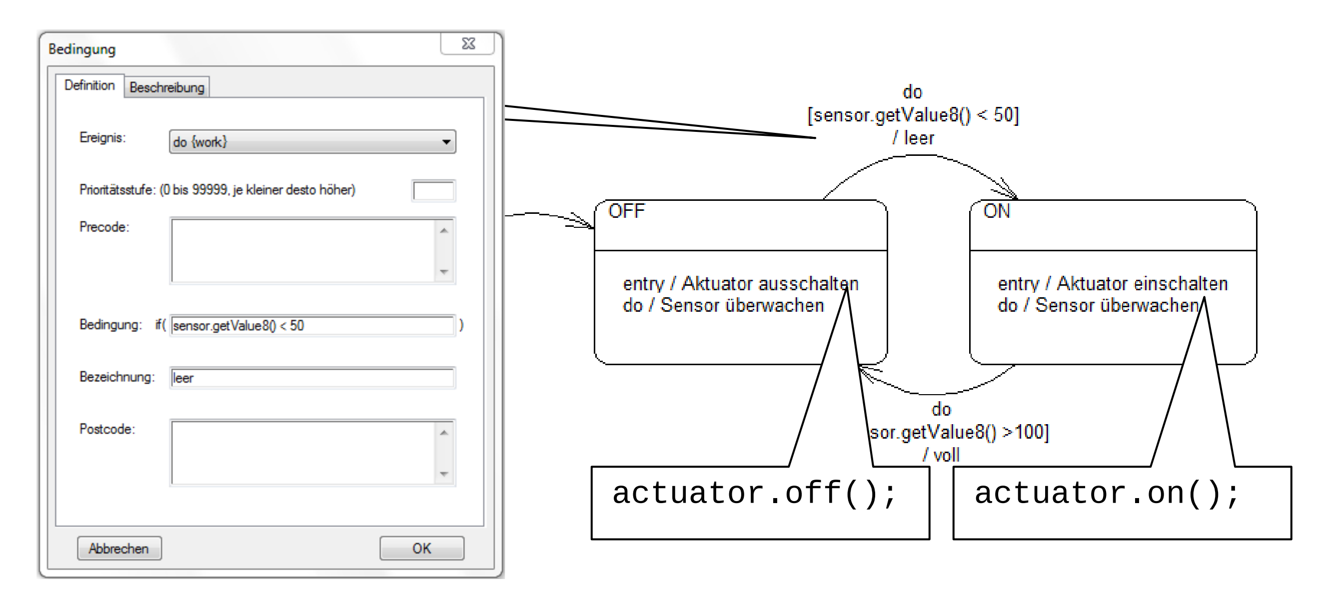
Danach werden die Entry-Aktivitäten beschriftet.



Um weitere Aktionen (entry/do/exit) einzufügen, müssen per Drag und Drop Objekte vom Typ Aktion in die betreffenden Zustände gezogen werden. Das Ereignis/Aktivitätstyp ist auszuwählen und die Aktivität zu beschriften.



Die Zustandsübergänge müssen definiert werden indem festgelegt wird, wann der Zustandswechsel erfolgen darf und unter welchen Bedingungen. Des Weiteren sollte der Zustandsübergang eine Bezeichnung erhalten.



Ist das Zustandsmodell erarbeitet, können die nötigen Statements für den Programmcode eingearbeitet werden. Dazu wird die betreffende Aktivität selektiert und im Quellcodeeditor die entsprechenden Befehle eingegeben. Danach kann im dazugehörigen Klassendiagramm (rechte Maustaste Menüpunkt *nach oben*) das Programm gebildet, übertragen und getestet werden.

11 Weitere Informationen zu SiSy

Versionen und Ausgaben Stand August 2012

Die Laufzeitkomponenten von SiSy, das Metamodell, Add-Ons sowie spezifische Skripte, Dokumente und Daten sind lizenzpflichtige Produkte. Ausgenommen sind zusätzlich bzw. nachträglich installierte Komponenten Dritter, die von SiSy als Modellierungswerkzeug lediglich zum Beispiel über Skripte angesprochen werden oder in die sich SiSy über offengelegte Schnittstellen eingebettet hat. Solche Produkte unterliegen den Lizenz und Nutzungsbedingungen des jeweiligen Herstellers bzw. Herausgebers.

Ausgaben

- SiSy Professional
  umfasst alle verfügbaren Add-Ons
- SiSy Business
  umfasst Add-Ons für Prozessmodellierung, Qualitäts- und Projektmanagement
- SiSy Developer
  umfasst umfangreiche Add-Ons zur Systementwicklung
- SiSy AVR ++
  umfasst die Add-Ons AVR, ausgewählte Teile der UML, SVL und SysML
- SiSy AVR
  beinhaltet nur das Add-On AVR
- SiSy ARM
  umfasst die Add-Ons ARM, ausgewählte Teile der UML und SysML

Lizenzmodelle

- Evaluation License
  erlaubt die Nutzung auf einem Arbeitsplatzrechner für Evaluierungszwecke
  Lizenzlaufzeit 1 Jahr
- Education License
  erlaubt die Nutzung auf einem Arbeitsplatzrechner für Ausbildungszwecke
  Vorraussetzung: Bildungseinrichtung, Student, Schüler, keine kommerzielle Nutzung
- Private License
  erlaubt die Nutzung auf einem Arbeitsplatz für private oder hausinterne Zwecke
  Vorraussetzung: ausschließlich für private, nicht kommerzielle Nutzung
  in Unternehmen hausinterne Nutzung der mit SiSy erstellten Produkte
- Single License
  erlaubt die Nutzung auf einem Arbeitsplatz kommerzielle Zwecke oder
  in öffentlichen Verwaltungen
- Enterprise License
  erlaubt die Nutzung auf bis zu 20 Arbeitsplatzen für kommerzielle Zwecke oder
  in öffentlichen Verwaltungen

Anhang: Tastaturbelegung, allgemein

Die Tastenbelegung ist abhängig vom jeweiligen Diagramm und der verwendeten Ausgabe:

F1	Hilfe wird geöffnet
F2	Zoomen
F4	Textfeld / Infofeld vergrößern
F5	Farbe bei Rahmen wird geändert; Form am Anfang einer Verbindung ändert sich
F6	Bei Rahmen und Verbindungen ändert sich die Form
F7	Bei Rahmen und Verbindungen ändert sich der Linientyp
F8	Form am Ende einer Verbindung ändert sich
F9	Bewirkt, dass der Mittelpunkt einer Kante auf Null gesetzt wird
ESC	Im Diagramm: Nach oben
	Im Dialog: Bricht ihn ab ohne zu speichern
Tab	In der Reihenfolge, in der die Objekte erstellt wurden, werden sie markiert
Umschalttaste + Enter	Objektbeschreibung wird geöffnet
Leertaste	Objektbeschreibung wird geöffnet
Alt + Enter	Dialog Definieren
Strg + Enter	Diagramm nach unten
Strg + \`R\`	Report für selektiertes Objekt
Strg + \`A\`	Executebefehl ausführen (nur in bestimmten Ausgaben)
Strg + \`D\`	Diagrammreport wird aufgerufen
Strg + \`I\`	Import von Diagrammen
Strg + \`T\`	Tokensteuerung starten/beenden
Strg + \`X\`	Export von Diagrammen
Strg + \`+\`	Selektiertes Objekt wird vergrößert (nur in bestimmten Ausgaben)
Strg + \`-\`	Selektiertes Objekt wird verkleinert (nur in bestimmten Ausgaben)
Strg + \`*\`	Ursprüngliche Objektgröße wird wiederhergestellt

Strg + Maustaste	Selektiertes Objekt wird innerhalb des Diagramms kopiert
Strg + Cursortasten:	
- Cursor nach links	Selektiertes Objekt wird in X-Richtung verkleinert
- Cursor nach rechts	Selektiertes Objekt wird in X-Richtung vergrößert
- Cursor nach oben	Selektiertes Objekt wird in Y-Richtung vergrößert
- Cursor nach unten	Selektiertes Objekt wird in Y-Richtung verkleinert
Enter	Editormodus zum Definieren der Objekte
Entf	Löschen
Cursortasten	Selektiertes Objekt wird verschoben (in Verbindung mit der Umschalttaste sind größere Schritte möglich)
\`+\`	Diagramm vergrößern
\`-\`	Diagramm verkleinern
\`*\`	Einpassen des Diagramms

Anhang: Mausoperationen

Die Maus hat in SiSy eine Anzahl von nützlichen Funktionen, welche die Arbeit in Projekten erleichtern.

Selektion	*Klick auf Objekt* Objekt ist markiert und kann separat weiterbearbeitet werden.
Selektion aufheben	*Klick auf Fensterhintergrund* Aufhebung der Objektmarkierung.
Mehrfachselektion	*Umschalttaste + Klick auf Objekt* Selektion/Markierung von mehreren Objekten zur Weiterbearbeitung. *Markise* Mit *Shift* und *gedrückter linker Maustaste* auf Fensterhintergrund und Ziehen eines Rechtecks über zu markierende Objekte.
Verschieben	*Drag & Drop im Diagramm* Objekt mit linker Maustaste anfassen und verschieben. Objekte werden am Raster verschoben. *Umschalttaste + Drag & Drop im Diagramm* Verschieben von Objekten ohne Raster.
Fensterinhalt schieben	*Linke und rechte Maustaste drücken + Verschieben der Maus im Diagramm* Der komplette Diagramminhalt wird geschoben.
Objekt kopieren	*STRG + Drag & Drop* Maustaste gedrückt halten und Mauszeiger vom Objekt auf den Fensterhintergrund führen. Eine Originalkopie des Objektes wird im aktuellen oder in einem anderen Diagramm erzeugt.
Referenz erzeugen	*Drag & Drop aus Navigator* Ziehen des gewünschten Objektes aus dem Navigator in das Diagramm. Es wird eine Referenz des gewählten Objektes erzeugt. *Drag & Drop aus Objektbibliothek* Rot beschriftete Objekte können nur als Referenz erzeugt werden. Eine Liste zur Auswahl des gewünschten Typs erscheint. *Strg + Drag & Drop aus Objektbibliothek* Eine Liste zur Auswahl der gewünschten Referenz des Originalobjektes erscheint.

	Drag & Drop aus anderem Diagramm Ziehen des gewünschten Objektes aus dem Quelldiagramm in das Zieldiagramm. Es wird eine Referenz des gewählten Objektes erzeugt.
Objekt anlegen	*Drag & Drop aus Objektbibliothek* Ein Objekt aus der Objektbibliothek wird im Diagramm angelegt und steht zur Verfeinerung bereit.
Objekt anhängen	*Drag & Drop Verteiler auf Fensterhintergrund* Durch Ziehen einer Kante vom Verteiler auf den Fensterhintergrund wird ein neues Objekt erzeugt. Nach Auswahl des Objekttyps sind die Objekte miteinander verbunden.
Objekte verbinden	*Drag & Drop Verteiler zu Objekt* Klick auf den Verteiler des zu verbindenden Objektes. Bei gedrückter linker Maustaste auf das gewählte Objekt ziehen. *Verbindung aus Objektbibliothek (in der UML)* Hierbei wird erst die gewünschte Verbindung in der Objektbibliothek angeklickt und danach die beiden zu verbindenden Objekte im Diagramm nacheinander.
Verbindung anordnen	*Drag & Drop Mittelpunkt* Beliebige Gestaltung der Verbindung durch Ziehen mit der Maus.
Verbindung ändern	*Drag & Drop Anfangs-/Endpunkt einer Kante* Für die Verbindung wird ein neues Zielobjekt gewählt.
Objekt definieren	*Doppelklick auf Objekt* Durch Doppelklick auf Objekte öffnet sich das Kontextmenü. Bei Abschalten des Menüs unter *Einstellungen*/*Menü bei Doppelklick* erscheint eine Zeile zur Namensgebung. Mit ESC wird die Eingabe bestätigt. *Doppelklick auf Verteiler* Es wird der Definieren-Dialog aufgerufen, in dem das Objekt benannt und beschrieben werden kann.
Kontextmenü öffnen	*Klick mit rechter Maustaste auf Objekt*
Fenster neu zeichnen	*Doppelklick auf Fensterhintergrund* *Hinweis*: Doppelklick mit linker Maustaste wirkt wie Enter.
Fenster aktualisieren	*Strg + Doppelklick auf Fensterhintergrund* Die vom Programm ausgeführten aber noch nicht sichtbargemachten Befehle werden im Fenster erstellt. Das Fenster wird aktualisiert.

Raum für eigene Notizen